목차

2021년 5월 29일 토요일 오후 5:14

번호	수업 내용
1	대화식 모드로 프로그래밍하기
2	텍스트 에디터로 프로그래밍하기
3	변수명 만들기
4	변수에 값 대입하기
5	주석 처리하기(#)
6	자료형 개념 배우기
7	자료형 출력 개념 배우기(print)
8	들여쓰기 개념 배우기
9	if문 개념 배우기 ① (if~else)
10	if문 개념 배우기 ② (if~elif)
11	for문 개념 배우기 ① (for)
12	for문 개념 배우기 ② (for~continue~break)
13	for문 개념 배우기 ③ (for~else)
14	while문 개념 배우기(while~continue~break)
15	None 개념 배우기
16	정수형 자료 이해하기
17	실수형 자료 이해하기
18	복소수형 자료 이해하기
19	대입 연산자 이해하기(=)
20	사칙 연산자 이해하기(, -, *, /, **)
21	연산자 축약 이해하기(=, -=, *=, /=)
22	True와 False 이해하기
23	관계 연산자 이해하기(==, !=, < , < =, > , > =)
24	논리 연산자 이해하기(and, or, not)
25	비트 연산자 이해하기(&, |, ~, ^, > > , < <)
26	시퀀스 자료형 이해하기
27	시퀀스 자료 인덱싱 이해하기

28	시퀀스 자료 슬라이싱 이해하기
29	시퀀스 자료 연결 이해하기( )
30	시퀀스 자료 반복 이해하기(*)
31	시퀀스 자료 크기 이해하기(len)
32	멤버체크 이해하기(in)
33	문자열 이해하기
34	문자열 포맷팅 이해하기
35	이스케이프 문자 이해하기
36	리스트 이해하기([ ])
37	튜플 이해하기(( ))
38	사전 이해하기({ })
39	함수 이해하기(def)
40	함수 인자 이해하기
41	지역변수와 전역변수 이해하기(global)
42	함수 리턴값 이해하기(return)
43	파이썬 모듈 이해하기
44	파이썬 패키지 이해하기
45	파이썬 모듈 임포트 이해하기 ① (import)
46	파이썬 모듈 임포트 이해하기 ② (from~import)
47	파이썬 모듈 임포트 이해하기 ③ (import~as)
48	파일 열고 닫기(open, close)
49	클래스 이해하기(class)
50	클래스 멤버와 인스턴스 멤버 이해하기
51	클래스 메소드 이해하기
52	클래스 생성자 이해하기
53	클래스 소멸자 이해하기
54	클래스 상속 이해하기
55	예외처리 이해하기 ① (try~except)
56	예외처리 이해하기 ② (try~except~else)
57	예외처리 이해하기 ③ (try~except~finally)
58	예외처리 이해하기 ④ (try~except Exception as e)
59	예외처리 이해하기 ⑤ (try~except 특정 예외)
60	사용자 입력받기(input)

61	자료형 확인하기(type)
62	나눗셈에서 나머지만 구하기(%)
63	몫과 나머지 구하기(divmod)
64	Pandas 를 이용한 데이터 검색
65	Pandas 를 이용한 조인
66	Pandas 를 이용한 서브쿼리
67	Pandas 와 오라클 그룹함수의 비교
68	반올림수 구하기(round)
69	실수형 자료를 정수형 자료로 변환하기(int)
70	정수형 자료를 실수형 자료로 변환하기(float)
71	정수 리스트에서 소수만 걸러내기(filter)
72	최대값, 최소값 구하기(max, min)
73	판다스에서 결측치(NaN) 확인하기
74	판다스에서 파생변수 추가하는 방법
75	문자열에서 특정 위치의 문자 얻기
76	문자열에서 지정한 구간의 문자열 얻기
77	문자열에서 홀수 번째 문자만 추출하기
78	문자열을 거꾸로 만들기
79	두 개의 문자열 합치기( )
80	문자열을 반복해서 새로운 문자열로 만들기(*)
81	문자열에서 특정 문자가 있는지 확인하기(in)
82	문자열에서 특정 문자열이 있는지 확인하기(in)
83	문자열 길이 구하기(len)
84	문자열이 알파벳인지 검사하기(isalpha)
85	문자열이 숫자인지 검사하기(isdigit)
86	문자열이 알파벳 또는 숫자인지 검사하기(isalnum)
87	문자열에서 대소문자 변환하기(upper, lower)
88	문자열에서 좌우 공백 제거하기(lstrip, rstrip, strip)
89	문자열에서 문자 개수 구하기 (count)
90	문자열에서 특정 문자(열) 위치 찾기 (find)
91	문자열을 특정 문자(열)로 분리하기(split)
92	문자열을 특정 문자(열)로 결합하기(join)
93	문자열에서 특정 문자(열)를 다른 문자(열)로 바꾸기(replace)

94	문자열을 바이트 객체로 바꾸기(encode)
95	바이트 객체를 문자열로 바꾸기(decode)
96	순차적인 정수 리스트 만들기(range)
97	리스트에서 특정 위치의 요소 얻기
98	리스트에서 특정 요소의 위치 구하기(index)
99	리스트에서 특정 위치의 요소를 변경하기
100	리스트에서 특정 구간에 있는 요소 추출하기
101	리스트에서 짝수 번째 요소만 추출하기
102	리스트 요소 순서를 역순으로 만들기 ① (reverse)
103	리스트 요소 순서를 역순으로 만들기 ② (reversed)
104	리스트 합치기( )
105	리스트 반복하기(*)
106	리스트에 요소 추가하기(append)
107	리스트의 특정 위치에 요소 삽입하기(insert)
108	리스트의 특정 위치의 요소 제거하기(del)
109	리스트에서 특정 요소 제거하기(remove)
110	리스트에서 특정 구간에 있는 모든 요소 제거하기
111	리스트에 있는 요소 개수 구하기(len)
112	리스트에서 특정 요소 개수 구하기(count)
113	리스트 제거하기(del)
114	리스트 요소 정렬하기 ① (sort)
115	리스트 요소 정렬하기 ② (sorted)
116	리스트 요소 무작위로 섞기(shuffle)
117	리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate)
118	리스트의 모든 요소의 합 구하기(sum)
119	리스트 요소가 모두 참인지 확인하기(all, any)
120	사전에 요소 추가하기
121	사전의 특정 요소값 변경하기
122	사전의 특정 요소 제거하기(del)
123	사전의 모든 요소 제거하기(clear)
124	사전에서 키만 추출하기(keys)
125	사전에서 값만 추출하기(values)
126	사전 요소를 모두 추출하기(items)20130 사전에 특정 키가 존재하는지 확인하기(in)

127	사전 정렬하기(sorted)
128	문자 코드값 구하기(ord)
129	코드값에 대응하는 문자 얻기(chr)
130	문자열로 된 식을 실행하기(eval)
131	이름없는 한줄짜리 함수 만들기(lambda)
132	인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map)
133	텍스트 파일을 읽고 출력하기(read)
134	텍스트 파일을 한줄씩 읽고 출력하기 ① (readline)
135	화면에서 사용자 입력을 받고 파일로 쓰기(write)
136	텍스트 파일에 한줄씩 쓰기(writelines)
137	텍스트 파일 복사하기(read, write)
138	바이너리 파일 복사하기(read, write)
139	파일을 열고 자동으로 닫기 (with ~ as)
140	HTML 기본문법
141	beautiful soup 모듈 배우기
142	웹스크롤링 실전 1단계 (ebs 레이디 버그 게시판)
143	웹스크롤링 실전2 (중앙일보사)
144	파이썬에서 워드 클라우드 그리기
145	웹스크롤링 실전3 (동아일보)
146	웹에 있는 사진을 스크롤링 하는 방법 (구글 이미지)
147	이미지 스크롤링 하기 (네이버 이미지 검색)
148	파이썬과 Oracle 연동
149	파이썬과 mySQL 연동
150	이미지를 숫자로 변환하는 방법 (폐사진)
151	이미지를 숫자로 변환하는 방법 (개와 고양이)
152	필수 알고리즘1 (합성곱 연산)
153	필수 알고리즘2 (이진 탐색)
154	필수 알고리즘3 (버블정렬)
155	필수 알고리즘4 (탐욕알고리즘)
156	필수 알고리즘5 (재귀알고리즘)
157	필수 알고리즘6 (LUR 알고리즘)
158	필수 알고리즘7 (자카드 유사도 알고리즘)
159	필수 알고리즘8 (비밀지도)

160	통계 마지막 문제 (도박사 이야기)

출처: <https://cafe.daum.net/_c21_/bbs_read?grpid=zchT&fldid=SfZF&contentval=00001zzzzzzzzzzzzzzzzzzzzzzzzz&datanum=1&page=1&prev_page=0&firstbbsdepth=&lastbbsdepth=zzzzzzzzzzzzzzzzzzzzzzzzzzzzzz&listnum=20>

11/24 (001-011)

2020년 11월 24일 화요일 오전 9:42

- 12기_파이썬 수업
- 데이터 분석가가 기본적으로 갖추어야 할 기술

1. SQL
2. 파이썬 + 통계 기본지식 + 머신러닝 지식
3. R 알고리즘 문제
4. 하둡

파이썬을 통해서 우리가 얻어야 할 지식?

1. 빅데이터 수집을 위한 웹스크롤링을 할 수 있는 능력 키우기
2. SQL처럼 파이썬으로 데이터를 검색하고 데이터를 분석할 수 있는 능력 키우기
3. 딥러닝 개발자(연구원)가 되기 위한 기본 코딩 능력 갖추기

- 파이썬 설치
  아나콘다 최신 버전으로 설치 (구글 -> 아나콘다 설치 검색)
  ↓
  파이썬 기본 프로그램 + 여러가지 유용한 패키지들

  spyder 툴로 파이썬 코드를 연습

- 파이썬으로 할 수 있는 것들?

1. 인공지능 ---> 머신러닝 ---> 신경망 ---> 딥러닝

- 스파이더(spyder) 프로그램 실행
- anaconda prompt --> python이라고 치기

  - 파이썬은 오라클과 다르게 실행 코드 끝에 ;를 붙이는게 아니라 다음 라인에 #%%을 붙여주어야 새로운 문장들을 각각 실행시킬 수 있다.
  - 들여쓰기는 tab이나 무조건 4번 띄워쓰기 해야한다.

■ 1. 대화식 모드로 프로그래밍하기

- 파이썬을 실행하는 방법 2가지

1. 대화식 모드 : 한라인 한라인씩 실행하는 모드
   Ex) Anaconda Prompt에서 한라인씩 실행한 모드
   >>> a=1
   >>> b=2
   >>> a+b
2. 배치 모드 : 여러개의 스크립트로 작성해서 한번에 실행하는 모드
   스파이더에서 아래와 같이 스크립트를 적어주고 전체 선택한 후에 ctrl + enter로 실행한 것
   a=1

```
b=2
print(a+b)
```

■ 2. 텍스트 에디터로 프로그래밍하기

- spyder와 같은 텍스트 창에서 프로그램 코드를 작성하는 것을 말한다.
- spyder 프로그램에서 ctrl + enter키로 코들를 실행한다.

■ 3. 변수명 만들기

" 어떤 값을 임시로 저장하는 변수의 이름을 만드는 방법과 규칙을 배웁니다"

Ex) a=1
설명 : a라는 변수에 숫자1을 할당한다.

※ 변수 이름을 생성할 때 주의할 사항

1. 변수 이름에는 다음 문자만 사용할 수 있습니다.
   a. 소문자 (a - z)
   b. 대문자 (A~Z)
   c. 숫자 (0~9)
   d. 언더스코어(_)

2. 변수 이름은 숫자로 시작할 수 없습니다. (문자로 시작해야 합니다.)
3. 예약어를 사용할 수 없습니다 (파이썬에서 이미 사용되고 있는 단어를 예약어라고 합니다.)
   Ex) True, False, class, is, return

■ 4. 변수에 값 대입하기

- 다양한 값을 변수에 대입하는 방법을 배웁니다.
- 파이썬은 변수에 값을 대입할 때 = (assignment)기호를 사용합니다. / 이퀄이 아님

Ex)

```
a = 1
b = 'scott'
print (b)
```

다른 프로그램 언어에서는 아래 같이 좀 더 복잡한 코드로 작성해야한다.

```
b varchar2 (10) ;= 'scott';
```

파이썬의 코드는 심플함을 철학으로 삼고 있습니다.
파이썬의 기본 철학을 확인하는 방법:

```
import this
```

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
```

Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

파이썬은 c/c++, java와는 달리 변수를 선언할 때 숫자형 자료인지 문자형 자료인지 자료형을 명시하지 않아도 됩니다.

PL/SQL	**vs**	**파이썬**
a number(10) : = 2;		**a=2**

■ 5. 주석 처리하기(#)

프로그램에서 주석부분은 인터프리터에 의해 무시되는 텍스트의 한 부분입니다. 코드를 설명하거나 나중에 어떤 문제를 고치기 위해 표시하는 등 다양한 목적으로 주석을 사용할 수 있습니다.
코드를 작성할 때 주석을 잘 작성해두면 차후에 코드를 다시 보거나 타인이 코드를 검토할 때 매우 중요한 정보로 활용이 될 수 있습니다. 그래서 주석을 항상 달아주는 습관을 가지는 것이 좋습니다.

Ex)

1. 한줄 주석

```
a = 1  #변수 a에 숫자 1을 할당합니다.
print(a)  #a의 결과를 출력합니다.
```

2. 여러줄 주석

```
""" 아래의 프로그램은 a 변수에 1을 할당해서 프린트하는 프로그램입니다."""
a = 1
print(a)
```

■ 6.자료형 개념 배우기

"자료형이란 프로그래밍을 할 때 쓰이는 숫자, 문자열등의 자료 형태로 사용되는 모든것을 뜻합니다."

- 파이썬에서 자주 다루게 되는 자료형이 5가지가 있습니다.

1. 숫자형 자료형 : 숫자를 표현하는 자료형
   Ex) a=1
2. 문자형 자료형 : 문자를 표현하는 자료형
   Ex) b='scott'
3. 리스트 자료형 : [ ] 대괄호 안에 임의 객체를 순서있게 나열한 자료형
   Ex) d = [1,2,3] #파이썬은 시작을 0부터 시작합니다.
   0 1 2
   print (d[0]) # d 리스트에서 첫번째 요소를 프린트해라 !
   파이썬은 0번부터 시작 /
4. 튜플 자료형 : 리스트와 비슷하지만 요소값을 변경할 수 없다는 것이 리스트와 다른점 입니다.
   Ex) c = (1, 2, 3)

print(c[0])
예를들면, 물건을 살 때 포인트 적립을 하게 되는데 총 가격의 10%만 적립이 된다고 하면 반드시 10%만 적립이 되어야 합니다. 30% 적립 노노!!
이럴 때는 소괄호를 사용해주어야 합니다.

5. 사전 자료형 : 사전 자료형은 { } 중괄호 안의 키 값으로 된 쌍이 요소로 구성된 순서가 없는 자료형 입니다.

Ex) m = { '|' : '나는'. 'am':'입니다', 'boy':'소년'}
print(m)

■ 7. 자료형 출력 개념 배우기(print)

print 함수를 이용하면 다양한 자료형을 화면에 출력할 수 있습니다.

Ex)
```
a = 200
b = 'i love python'
c = ['a', 'b', 'c']
print (a)
print(b)
print (c)
```

■ 8 들여쓰기 개념 배우기

파이썬에는 실행코드 부분을 묶어주는 ( ) 괄호가 없습니다.
대신 들여쓰기로 괄호를 대신합니다.
파이썬은 다른 프로그래밍 언어와 달리 if, for, while 등과 같은 제어문, 루프문의 실행코드 부분을 구분해주는 괄호가 없습니다.
파이썬에서 제어문이나 함수이름, 클래스 이름 뒤에 콜론(:)으로 제어문, 함수이름, 클래스 이름의 끝을 표시하며 콜론(:) 다음에 실행코드를 작성하는데 이때 들여쓰기를 해야합니다.

Ex 1) 실행코드를 다음라인에 작성했을 경우

```
listdata = ['a', 'b', 'c']                  # listdata라는 변수로 리스트를 작성
if 'a' in listdata :                        # listdata 변수에 'a'요소가 있다면 아래의
    print ('a가 listdata에 있습니다.')      #실행문을 실행해라~~
```

Ex 2) 실행코드를 한 라인에 작성한 경우
```
listdata =['a','b','c']
if 'a' in listdata : print ('a가 listdata에 있습니다.')
```

※ 설명 : 콜론 (:) 으로 if문의 끝을 알린다.

■ 9. if문 개념 배우기 ① (if~else)

어떤 조건을 참과 거짓으로 판단할 때 if문을 사용합니다.
참과 거짓을 구분하여 코드를 실행하면 if ~ else를 사용합니다.
코드를 작성하다 보면 조건에 따라 수행하는 일을 달리 해야하는 경우가 있습니다.
조건이 참인지 거짓인지 검사를 하고 참인 경우에는 이 일을 하고 거짓인 경우에는 저 일을 해라 라는 식으로

처리를 할 수 있습니다.

Ex) if 조건:
    실행코드1
  else:
    실행코드2

Ex) a = int(input ('숫자를 입력하세요~'))
  if a%2 == 0: #%는 나눈 나머지값을 출력하는 연산자
    print ('짝수 입니다') # == 은 equal 연산자 입니다.
  else: # = 은 할당 연산자 입니다.
    print('홀수 입니다')

  == : 이퀄(equal)

■ 10. if문 개념 배우기 ② (if~elif)

여러개의 조건을 순차적으로 체크하고 해당 조건이 참이면 특정 로직을 수행하고자 할 때 if ~ elif문을 사용합니다.

Ex)
  if 조건1:
    실행코드1
  elif 조건2:
    실행코드2
  elif 조건3:
    실행코드3
  else 조건4: #위의 조건1, 조건2와 조건3이 아니라면
    실행코드4

- **문제 13번을 다시 수행하는데 같은 숫자가 2개 들어오면 서로 같습니다. 라는 메세지가 출력되게 하시오.**

a = int(input ('첫번째 숫자를 입력하세요~'))
b = int(input ('두번째 숫자를 입력하세요~'))
c =
if a<b :
  print (a,'는 ',b,'보다 작습니다.')
elif a>b:
  print (a,'는 ', b, '보다 큽니다.')
else:
  print (a,'는 ', b,'와 같습니다')

■11. for문 개념 배우기 ① (for)

특정 코드를 반복적으로 수행하기 위해서는 반복문을 사용해야 하는데 파이썬에서는 for문이 반복문을 수행하기 위해 가장 많이 사용되는 문법입니다.

Ex) for 변수 in 범위:

반복적으로 실행할 코드

1. 리스트 범위인 경우:

```
for i in [1,2,3]:
        print(i)
```

2. 튜플 범위인 경우:

```
for i in (1,2,3):
        print(i)
```

3. range ( ) 범위인 경우:

```
for i in range(10):
    print (i)
```

```
>>>
0
1
2
3
4
5
6
7
8
9
```

0번~9까지 출력해서 10개가 나온다.

4. 사전형 범위인 경우

```
m = {'i':'나는', 'am':'입니다', 'boy':'소년'}
for i in m:
        print(i)          #키값만 출력되고 있습니다.
```

```
>>>
i
am
boy
```

5. 문자형 범위인 경우

```
for i in 'i am a boy':
        print(i)
```

```
>>>
i

a
m

a

b
o
y
```

■ for 루프(loop)문의 range 사용법 정리

```
for i in range(1,6):
    print (i) #1부터 5까지 출력한다.
```

>>> 마지막 숫자 하나 전까지 출력함

```
for i in range(1,6,2):
    print (i)                    #1,3,5를 출력한다.

for i in range(6,1,-1):
    print(i)                     #6부터 1씩 차감해서 2까지 출력한다.
```

11/25 (012-018)

2020년 11월 25일 수요일 오전 9:44

■ 데이터 분석가 및 딥러닝 개발자 되기 위해서 갖추어야할 기술

1. SQL
2. 파이썬

■ 중첩 for loop문

loop문을 중첩시켜서 수행한다. loop문 자체를 반복시킨다.

Ex) for j in range(1,10) :
    for i in range(1,10):
        print('2 x ', i, '=', 2*i)
    설명 : 2단을 출력하는 for loop문이 9번 실행되면서 2단이 9번 나왔습니다.

2단을 9번 나오게 하는게 아니라 2단부터 9단까지 출력해야하므로 다음과 같이 작성해야 합니다.

```
for j in range(2,10) : #j는 2부터 10미만까지 출력이 된다.
    for i in range(1,10): #i는 1부터 10미만까지 출력이 된다.
        print( j ' x ', i, '=', j*i)
```

설명 : j = 2일때 i가 1~9까지 반복 실행

```
2 x 1 = 2
2 x 2 = 4
  :
  :
2 x 9 = 18
```

j = 3 일때 i가 1~9까지 반복 실행

```
3 x 1 = 3
3 x 2 = 6
  :
  :
3 x9 = 27
```

j = 4 일때 i가 1~9까지 반복 실행

```
:
:
```

j = 9일때 i가 1~9까지 반복 실행

- 리스트 append함수의 기본 문법

a = [ ] #비어있는 리스트 a를 생성합니다.
a.append(7) #a리스트에 숫자7을 넣는다.
print(a)
a.append(8) #a리스트에 숫자8을 넣는다.

print(a)

■ 12. for문 개념 배우기 ② (for~continue~break)

for 반복문 내에서 continue를 만나면 그 다음 반복 실행으로 넘어가며 break를 만나면 for 반복문을 완전히 벗어나게 됩니다.

Ex) for 변수 in 범위:
....실행코드
continue #이 부분은 그냥 지나치고 다음 반복문을 수행해라~
.... 실행코드
break #for 반복문을 탈출

■ for문에서 사용하는 break 문
" 루프문을 중단시키는 역할을 하는 키워드"

Ex) i in range(1,11) : #1부터 10까지 루프를 돌리는데
print (i) #i를 출력을 하다가
if i == 3: #i가 3잉면
break #루프문을 종료시켜라~

■ 13. for문 개념 배우기 ③ (for~else)
for 반복문을 완전히 수행했을때만 실행하는 부분을 정의하려면 for~else문을 사용해야 합니다.
for~else에서 else뒤에 실행되는 코드는 for 반복문을 성공적으로 수행해야지만 실행 됩니다.
Ex)

```
for i in range(1,11):
    print(i)
else:
    print('perfect') # break에 의해서 루프문이 중단되면 else다음에
                     # 실행문이 실행되지 않습니다.
>>>
1
2
3
4
5
6
7
8
9
10
perfect
```

■ 14. while문 개념 배우기(while~continue~break)

for 루프문처럼 while루프문도 같은 반복문 입니다.
for 루프문은 특정 범위에서 반복 실행하게 하는 반면에 while 루프문은 특정 조건에서 코드를 반복 실행하게 합니다.

```
for 루프문                    vs          while루프문
for i in range(0,10):                      x=0
    print(i)                               while x <10:
                                               print(x)
                                               x = x+1

                                           0<10 : 참
                                           1<10 : 참
                                             :
                                             :
                                           9<10 : 참
                                           10<10: 거짓
```

■15. None 개념 배우기
None은 아무것도 없다는 의미의 상수 입니다.
아무것도 없다는 것을 나타내기 위해서 주로 활용됩니다.

```
Ex )
x = 0
while x < 10:
    print(x)
    x=x=1
```

위의 while loop문에서의 x변수는 숫자만 담기는 변수입니다.
그런데 어떨때는 숫자가 들어갈 수도 있고 어떨때는 문자가 들어갈수도 있어서 처음에 변수 선언시 결정을 못하겠다면 아래와 같이 하면 됩니다.

```
x = None #N은 대문자
a=1
if a==1:
   x=[1,2,3] #x라는 변수에 리스트를 담았습니다.
else:
   x='I love Phthon' #x라는 변수에 문자열을 담았습니다.
print(x)
```

■16. 정수형 자료 이해하기
자연수(1,2,3,....) 와 음수(-1,-2,-3,....)와 0으로 이루어진 수의 체계를 정수라고 합니다.

```
Ex ) a =123
     b = -178
```

```
    c = 0
```

일반 프로그래밍에서 지원하는 정수형 상수의 범위는
-2,147,473,647 ~2,147,473,647 인데 (2의 31승)
파이썬은 메모리가 허용하는 범위에서 지원 가능한 수를 다 사용할 수 있습니다.

```
for i in range(1000000000000000000000000000000000000000000000000):
    a=i
    print(i)
```

```
import random
print(random.uniform(0,1)) #0~1사이의 난수를 출력
```

■17. 실수형 자료 이해하기
실수는 소수로 나타낼 수 있는 유리수와 소수로 나타낼 수 없는 무리수로 구성된 집합입니다
(무리수의 예 : 원주율, 오일러 자연상수 e)
파이썬은 실수를 지원하기 위해서 부동 소수형을 제공합니다.

- 부동 소수형의 특징

1. 8바이트만 이용해서 수를 표현합니다. 즉 한정된 범위의 수만 표현할 수 있습니다.
2. 디지털 방식으로 소수를 표현해야 하므로 정밀도의 한계가 있습니다.

Ex) 소수 43.2에서 소수 43.1을 빼면 얼마인지 파이썬으로 구현하시오!

```
print(43.2-43.1)
>>>
0.10000000000000142
```

설명 : 부동 소수형은 위와 같이 정밀도의 한계를 가지고 있습니다. 이것은 비단 파이썬만의 문제가 아니라 부동 소수형이 어떻게 구성되고 동작되는가를 정의하는 표준인 IEEE 754를 따르는 모든 컴퓨터 시스템의 문제입니다. IEEE754는 제한된 메모리를 이용해서 (파이썬의 경우는 8바이트) 실수를 표현하기 때문에 제한된 정밀도를 갖습니다.

■18. 복소수형 자료 이해하기
복소수 :

1. **실수:**
   정수, 유리수, 무리수
2. **허수:**
   i를 제곱하면 -1이되는 수

복소수는 실수부와 허수부로 되어있고 허수부는 숫자 뒤에 문자 i를 이용하는데 파이썬에서는 j를 사용합니다.

Ex) c1 = 1 + 7j
실수부 허수부

```
c1 = 1+ 7j
print(c1.real) #복소수형 자료에서 실수부만 취한다.
print(c1.imag) # 복소수형 자료에서 허수부만 취한다.
c2=complex(2,3) #실수부가 2이고 허수부가 3인 복소수를 만든다.
print(c2)
>>>
(2+3j) >>>> 파이썬은 j가 허수
```

11/26 (019-025)

2020년 11월 26일 목요일 오전 9:43

■ 19. 대입 연산자 이해하기(=)

변수에 값을 대입하는데 사용하는 기호는 = 입니다.
수학에서 = 은 = 기준 왼쪽과 오른쪽의 값이 같다는 의미를 가지지만 파이썬을 포함한 컴퓨터 프로그래밍 언어에서 =은 = 왼쪽 변수에 = 오른쪽의 값을 대입하겠다는 의미입니다.

Ex) a = 7

- 대입 연산자와 일반 연산자를 비교

1. = : 대입 연산자
2. == : 같다
3. in : 여러개의 값을 비교

■ 20. 사칙 연산자 이해하기(, -, *, /, **)

- 사칙 연산자를 파이썬과 오라클과 비교

오라클	파이썬	
+	+	
-	-	
*	*	
/	/	
mod	%	
power : 2^3	power(2,3) , 2**3	
sqrt	sqrt	
log2	log2	밑수가 2인 로그함수
log10	log10	밑수가 10인 로그함수

Ex 1) 밑수가 10이고 진수가 10인 log값을 출력하시오.
(log 함수를 사용하려면 math모듈을 import 해야합니다.)

import math #수학에 관련한 함수들이 다 코딩되어 있는 모듈 math를 이 코드에서 사용하겠다.

print(math.log10(10))
진수값
모듈이름.함수이름(매개변수 값)

■ 21. 연산자 축약 이해하기(=, -=, *=, /=)

변수에 값을 사칙 연산하여 그 결과를 동일한 변수에 대입할 때 연산자를 축약해서 대입할 수 있습니다.

Ex) cnt = cnt+1 를 다음과 같이 쓸 수 있습니다.

cnt += 1

축약

Ex) cnt = cnt+1 ------> cnt += 1과 같다.

cnt = cnt -1 ------> cnt -= 1과 같다.

cnt = cnt * 2 -----> cnt *= 2와 같다.

cnt = cnt / 4 -----> cnt/= 4 와 같다.

■ 22. True와 False 이해하기

참을 나타내는 것이 True이고 거짓을 나타내는 것이 False입니다.
조건을 판단해서 그 조건이 참이면 True, 거짓이면 False를 리턴합니다.

True는 1, False는 0 값을 가집니다. 참과 거짓을 나타낼때는 True와 False로 표현하면 더 직관적이고 프로그램 코드 가독성을 높일 수 있습니다.

```
Ex ) a = True
     b = False
     print (a)
     print ( 1==1)
     ----------
     a= True
     b = False
     print(a)
     print (1==1) #결과가 True로 출력됩니다.
     print(1==2) #결과가 False로 출력됩니다.
```

■23. 관계 연산자 이해하기(==, !=, < , < =, > , > =)

오라클	vs	파이썬
>		>
>=		>=
<		<
<=		<=
=		==
!=		!=
in		in

**

```
print(np.mean(a)) #평균
print(np.var(a)) #분산
print(np.std(a)) #표준편차
```

■25. 판다스(pandas) 사용법

- Pandas란 ? 1. 데이터 분석을 위한 파이썬 모듈
  2. 엑셀의 스프레드 시트와 같은 관계형 데이터베이스의 데이터 처리 능력이 뛰어남
  3. 판다스는 Dataframe이라는 기본 자료구조를 사용합니다.
     (오라클의 테이블과 판다스의 Dataframe이 서로 유사합니다.)

- 지금 카페에 올린 emp3.csv 와 dept3.csv를 c 드라이브 밑에 data라는 폴더를 만들고 그 밑에 복사해 둡니다.

```
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")    #c드라이브 밑에 emp3.csv를 읽어서 emp 변수에
print(emp)                                    #넣는다
```

설명 : 판다스의 read_csv함수는 os의 csv파일을 읽어서 파이썬에서 판다스 데이터 프레임으로 만들겠다.

SQL ---> 데이터에서 정보를 얻어내기 위해 (데이터 검색)
파이썬의 판다스 ---> 데이터를 검색하는 모듈

■ 오라클의 기타 비교 연산자와 파이썬 판다스와 비교

	오라클	vs	판다스
1.	between.. and		emp['sal'].between(1000,3000)
2.	in		emp['job'].isin(['SALESMAN', 'ANALYST'])
3.	is null		emp['comm'].isnull()
4.	like		apply 함수

11/30 (026-034)

2020년 11월 30일 월요일 오전 9:50

■ 복습

"파이썬은 괄호 대신에 들여쓰기로 블럭을 구분한다"

Ex) for i in range(1,11):
prit('a') #<---- for loop 문의 관할 하에 ㅣ ㅆ는 실행영역
print('b') #<-- for loop문의 관할하에 있는 실행영역이 아닙니다.
for loop문이 다 끝나고 나면 작동되는 문장

SQL과 파이썬의 차이점?

1. SQL은 비절차적 언어 : select ename, sal
from emp
where ename = 'SCOTT'
2. 파이썬은 절차적 언어 : 코드의 흐름대로 절차적으로 실행한다.

```
for i in range(1,10):
    print('a')
print('b')
for k in range(1,100):
    print('c')
```

1. 파이썬 기본 자료형 5가지
   a. 문자형 : a = 'scott'
   b. 숫자형 : cnt = 0 ★
   c. 리스트형 : a = [1,2,3] 대괄호 ★
   d. 튜플 : b = (1,2,3) 소괄호
   e. 딕셔너리 : c = {'apple' : '사과', 'eye', : '눈'} 중괄호
   키 값
2. 파이썬 연산자 : >, < , >=, <=, ==, !=
3. if문 : if조건 :
   실행문
4. loop문 : 프로그램을 실행하다 보면 반복적으로 특정 실행문을 실행해야 할 때가 있습니다.
   그 때 loop문을 사용해야 합니다.
   for loop문
   Ex) for i in range(1,11):
   Print('a') i=1 일때 print('a')
   i=2일때 print('a')

:
i=10일때 print('a')

while loop문

5. 판다스 사용법 : 판다스는 엑셀과 같은 데이터를 검색할 때 주로 사용

database ----------파이썬 판다스
↓
emp 테이블 검색
↓ 데이터 시각화
SQL 통계를 쉽게 구현
머신러닝 쉽게 구현

"현업에서 주로 이렇게 해서 분석을 많이 합니다."
아직까지는 DATABASE와 연동을 하지는 않았고 emp.csv를 판다스로 읽어들여서 검색을 했습니다.

■ 26. 시퀀스 자료형 이해하기

시퀀스 자료형은 어떤 객체가 순서를 가지고 나열되어 있는 것을 말합니다. 예를 들어 문자열 'abcd'는 문자 a,b,c,가 순서를 가지고 차례대로 나열되어 있는 것입니다.

```
Ex )
a = 'scott'                    # s c o t t
print(a) #scott                  0 1 2 3 4
```

위의 scott이라는 문자를 담은 a변수에서 첫번째 요소만 출력하고 싶다면? --> s자만 출력

```
print (a[0] ) # s
print( a[1] ) # c
```

■ 27. 시퀀스 자료 인덱싱 이해하기

인덱싱(indexing)이란 시퀀스 자료형에서 인덱스를 통해 해당하는 값을 얻는 방법입니다. 파이썬에서는 인덱스를 0부터 시작하며 음수인 인덱스도 사용가능합니다. 음수 인덱스는 '끝에서 몇번째'라는 의미를 갖습니다.

```
Ex ) a = 'SALESMAN'
       print (a[2]) #L이 출력됨
```

■ 28. 시퀀스 자료 슬라이싱 이해하기

인덱싱은 인덱스에 해당하는 요소 하나를 취하는 방법이지만 슬라이싱은 시퀀스 자료에서

일정 범위에 해당하는 부분을 취하는 방법입니다.

Ex) a = 'scott'
print(a[1:3]) #co

s c o t t
0 1 2 3 4

설명 : 1번째자리부터 3미만의 자리까지 요소들을 출력

■ 29. 시퀀스 자료 연결 이해하기(+)

자료형이 동일한 두개의 시퀀스 자료는 + 연산자로 순서있게 연결하여 새로운 시퀀스 자료를 만들 수 있습니다. 문자열 + 문자열, 리스트 + 리스트, 튜플 + 튜플과 같이 두개의 동일한 시퀀스 자료형에 대해 '+'연산자로 연결이 가능합니다.

Ex)
a = 'i love '
b = 'python'
print(a+b)

■ 30. 시퀀스 자료 반복 이해하기(*)

동일한 시퀀스 자료를 반복하여 새로운 시퀀스 자료로 만들고자하면 별표(*)를 연산자로 사용합니다.

Ex) print('a'*7) #소문자 알파벳 a를 7번 출력합니다.
결과 : aaaaaaa

print([1,2,3]*5) #리스트 [1,2,3]안의 요소를 5개로 늘려서 출력

결과 : [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]

■ 31. 시퀀스 자료 크기 이해하기(len)

모든 시퀀스 자료는 고정된 길이 또는 크기를 가지고 있습니다.
시퀀스 자료의 크기는 시퀀스 자료를 구성하는 요소의 개수입니다.

Ex) a= 'scott'
print(len(a)) #5

b = [2,3,4,8,1]
print(len(b)) #리스트의 요소의 개수가 출력됩니다.

■ 32. 멤버체크 이해하기(in)

in은 자료에 어떤 값이 있는지 없는지 확인할 때 사용하는 키워드입니다.
활용하는 방법은 다음과 같습니다.

Ex) 값 in 자료
설명: 값이 자료에 있으면 True이고 없으면 False가 됩니다.

Ex)
```
listdata = [1,2,3,4]
a = 5 in listdata
b = 4 in listdata
print(a) #False
print(b) #True
```

■ 33. 문자열 이해하기

문자열은 문자나 기호 순서로 나열되어 있는 시퀀스 자료 입니다.

※ 문자열을 선언하는 방법 3가지

1. '문자열' <----- 싱글 쿼테이션 마크
2. "문자열" <----- 더블 쿼테이션 마크 한개
3. """문자열""" <----- 더블 쿼테이션 마크 3개

문자에 아래와 같이 싱글 쿼테이션 마크(')가 포함되어져 있으면 더블 쿼테이션 마크를 사용하면 됩니다.

```
a = "My son's name is John"
print(a)
```

문자에 아래와 같이 싱글 쿼테이션 마크와 더블 쿼테이션 마크가 있으면 3가지 방법 중 세번째인 """ """ 더블쿼테이션 마크를 3개 사용하면 됩니다.

```
b = """ My son's name is "jone" """
print(b)
```

■ 34. 문자열 포맷팅 이해하기

문자열 포멧팅이란 변하는 값을 포함하는 문자열을 표현하기 위해 하나의 양식으로 문자열을 만드는 것 입니다.
문자열 포멧팅에서는 변하는 값을 나타내기 위해 사용하는 기호를 '포멧 문자열은 다음과 같습니다.

포멧 문자열	설명
%s	문자열에 대응됨
%c	문자나 기호 한개에 대응됨
%f	실수에 대응됨
%d	정수에 대응됨
%%	%라는 기호 자체를 표시함

Ex)

```
txt1 = '자바'
txt2 = '파이썬'
num1 = 5
num2 = 10
print('나는 %s 보다 %s에 더 익숙합니다' %(txt1, txt2))


import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

for i in range(1,101):
        a.append(np.random.choice(mogipdan, 49).mean ())
print('표본평균의 평균값은 %.2f이고 분산값은 %.2f이고 표준편차 값은 %.2f입니다'
%(round(np.mean(a),2), round(np.var(a),2), round(np.std(a),2) ))
```

>>> .2f 이렇게 써주니까 소수점 2번째 자리까지 나왔다.

12/1 (035-043)

2020년 12월 1일 화요일 오전 9:36

■ 파이썬 복습

1. 파이썬의 자료형 5가지 : 문자형, 숫자형, 리스트형, 튜플형, 사전형
2. 파이썬의 연산자 : >, <, >=, <=, =, !=
3. 파이썬에서의 들여쓰기의 용도와 중요성
   자바나 c언어는 괄호로 특정 블럭을 지정하는데 파이썬은 들여쓰기로 특정 블록(실행영역)을 지정합니다.
   Ex)
   ```
   for i in range(1,11)
       print('a') # for loop문의 관할 하에 있는 실행영역
   print('b')      # for loop문과 상관없는 별도의 독립적인 실행영역
   ```
4. if문 사용법

   ```
   if 조건:
       실행조건
   elif 조건:
       실행코드
   elif 조건:
       실행코드
   else:
       실행코드
   ```

5. 반복문 : for loop문, while loop문

6. 판다스 사용법 : 데이터 검색, 시각화, 통계와 머신러닝 구현을 쉽게 해주는 파이썬의 모듈

7. 문자열 변수에서 특정 철자를 취하는 방법(인덱싱)
   ```
   a = 'scott'
   print(a[0])
   ```

8. 문자열 변수에서 특정 철자들을 취하는 방법(슬라이싱)
   ```
   a = 'scott'
   print(a[0:2]) #sc
   ```

9. 문자열 변수의 길이 확인하는 방법(len)
   ```
   a = 'scott'
   print(len(a)) #5
   ```

10. 멤버 체크 하는 방법(in)
    ```
    a=[1,2,3,4,5]
    b= 3 in a
    print(b) #true
    ```

11. 문자열 포멧 이해하기

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

for i in range(1,101):
        a.append(np.random.choice(mogipdan, 49).mean ())
print('표본평균의 평균값은 %.2f이고 분산값은 %.2f이고 표준편차 값은 %.2f입니다'
%(round(np.mean(a),2), round(np.var(a),2), round(np.std(a),2) ))
```

표준평균의 평균값은 30.06이고 분산값은 0.78이고 표준편차 값은 0.89입니다'

■ 35. 이스케이프 문자 이해하기

이스케이프 문자는 키보드로 입력하기 어려운 기호를 나타내기 위해 역슬래쉬 '₩'로 시작하는 문자입니다.
파이썬에서 자주 사용되는 이스케이프 문자는 다음과 같습니다.

이스케이프 문자	설명
₩n	줄바꾸기
₩t	텝
₩엔터	줄 계속
₩₩	₩ 기호자체
₩' 또는 ₩"	'기호 또는 "기호 자체

예제 : print('나는 파이썬을 배웁니다. ₩n파이썬은 자바보다 ₩ #여기서 엔터치기(줄바꿈)
훨씬 쉽습니다')

■ 36. 리스트 이해하기([ ])

리스트는 파이썬에서 가장 많이 활용되는 시퀀스 자료형 중 하나입니다.
리스트는 [ ]로 표시하며 [ ] 안의 요소를 콤마(,)로 구분하여 순서있게 나열합니다.

예제) k = ['a', 'b', 'c', 'd', 'e']
0 1 2 3 4

```
print(k)
print(type(k))              #<class 'list'>
print( k[1])                # b
```

■ 37. 튜플 이해하기(( ))

튜플은 리스트와 비슷한 성질을 가지고 있는 자료형이지만 요소의 값을 변경할 수 없다는 특징이 있습니다.
리스트는 대괄호 [ ]로 요소들을 감쌌는데 튜플은 소괄호 ( )로 요소를 감쌉니다.
튜플은 데이터가 변경이 안되므로 튜플로 만든 데이터에 대한 신뢰도가 높아집니다.

예제 1) 리스트에서 특정 요소를 변경하는 방법

```
a = [1, 2, 3, 4, 5]
print( type(a) )          #<class 'list'>
print( a[0] )             #1
a[0] = 7                  #a리스트의 0번째 요소를 7로 변경한다.
print(a)                  #[7, 2, 3, 4, 5]
                            ↑
                           a 리스트의 0번째 요소가 7로 변경되었습니다.
```

예제 2) 튜플은 데이터를 변경할 수 없습니다.

```
b = (1, 2, 3, 4, 5)
print( b[0] ) #1
b[0] = 7 #에러가 나면서 실행되지 않습니다.   'tuple' object does not support item assignment
```

설명 : 회사에서 변경이 되어서는 안되는 데이터는 프로그래밍할 때 튜플로 만들어서 관리하면 됩니다.

예) 신세계 백화점, 이마트에서 사용하는 포인트 적립 카드에서 포인트 적립 시 구매금액의 0.01%로 적립을 해준다고 하면 이 0.01은 절대로 프로그램에서 변경이 되어서는 안되는 데이터이므로 튜플로 관리해야합니다.

■ 38. 사전 이해하기({ })

사전은 키와 값을 하나의 요소로 하는 순서가 없는 집합입니다.
그러므로 사전은 시퀀스 자료형이 아니며 인덱싱으로 값을 접근 할 수도 없습니다. 사전은 '키:값' 쌍이 하나의 요소입니다.

예) a = {'apple':'사과', 'banana': '바나나', 'peach':'복숭아', 'pear':'배'}

```
print(a)
print(type(a)) #<class 'dict'>
print( a.keys( ) ) #키값들만 출력
print ( a.values ( ) ) #값들만 출력
a['grape'] = '포도' #새로운 키와 값을 추가하는 방법
print(a)
```

{'apple': '사과', 'banana': '바나나', 'peach': '복숭아', 'pear': '배', 'grape': '포도'}
이렇게 나온다.

예제) b = { } #비어있는 딕셔너리를 생성한다.

```
b['apple'] = '사과'
    ↑          ↑
   키         값
b['pear'] = '배'
b['grape'] = '포도'
print(b)

b = { }
```

■ 39. 함수 이해하기(def)

오라클 배울 때 함수를 다음과 같이 사용했었습니다.

```
select max(sal)
    from emp;

select substr('smith', 1, 3)
    from dual;
```

오라클 배울때는 오라클 개발자가 만들어준 함수를 사용했지만 파이썬 배울때는
우리가 직접 함수를 생성합니다.
파이썬에도 파이썬 개발자가 미리 만들어 둔 함수가 있습니다.
파이썬이 무료이고 전세계인들이 사용하는 프로그램이다 보니 각국의 뛰어난 개발자들이 파이선 함수를 직접 만들어서 공유해 줍니다.
우리는 그냥 import 명령어로 불러와서 그 개발자가 만든 함수를 사용할 수 있습니다.

오라클 함수	vs	파이썬 함수
upper		upper( )
lower		lower( )
initcap		만들어써야함
substr		만들어써야함
replace		replace( )
length		len( )
rtrim		rstrip( )
ltrim		lstrip( )
rpad		만들어써야함
lpad		만들어써야함
round		round( )
trunc		trunc( )
mod		%
power		pow
to_char		str
to_number		int, float
to_date		datetime, strptime( )
nvl		만들어써야함

decode if문

case if문

예제) 아래의 SQL을 파이썬으로 구현하시오!

```
SQL > select lower(ename)
            from emp;
```

Pandas>

```
#1
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
        print(type(i))
```

```
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
```

```
#2
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
        print(i.lower())
```

**** 함수를 생성하는 방법**

```
def 함수이름(매개변수):
        실행문
        return 출력값이 있는 변수명
```

예제:

- 두개의 입력값을 받아서 더하는 함수를 생성

```
 def add_number(n1, n2):
        result = n1 + n2  # n1에 입력된 값과 n2에 입력된 값을 더해서 result에 입력한다.
        return result         # result에 입력된 값을 출력한다.
```

n1, n2 --> 입력 매개변수

- 위에서 만든 함수를 실행하는 방법

```
print(add_number(1,2)) #3
```

설명 : 위와 같이 만들고 컴퓨터를 껐다가 켠다음 다시 아래와 같이 함수를 실행만 하게 되면 실행되지 않는다. 다시 add_number함수를 생성해줘야 합니다.

다시 아래와 같이 add_number함수를 생성하고 실행해줘야 합니다.

★ 콜론(:)으로 끝을 맺는 경우 4가지

1. if문 종료시
   예 : if 조건:

2. for문 종료시
   예 : for i in range(1,11):

3. 함수 생성시
   예 : def 함수이름(입력 매개변수):

4. 클래스 생성시
   예 : class 클래스이름:

■ 40. 함수 인자 이해하기

인자란 위에서 사용한 입력 매개변수를 말합니다.

예제:

```
def add_text( t1, t2 ):
    return t1 + ' ' + t2

print (add_text('파이썬', '자바'))
```

결과: 파이썬 자바

***매개변수에 아무것도 입력하지 않고 실행하면 기본값이 출력되게끔 함수를 생성하는 방법**

```
def add_text(t1, t2='최고'):     #t2 : 최고 -> 기본값
    return t1 + ' ' + t2

print(add_text('파이썬', '자바') ) #파이썬 자바
print( add_text('파이썬'))          #파이썬 최고
```

설명 : t2에 값을 아무것도 안 넣었더니 기본값으로 지정한 최고가 출력되었습니다.

■ 41. 지역변수와 전역변수 이해하기(global)

변수는 자신이 생성된 범위(코드블럭) 안에서만 유효하다.
함수 안에서 만든 변수는 함수 안에서 살아 있다가 함수 코드의 실행이 종료되면 그 생명을 다한다.
이것을 '지역변수'라고 합니다.

예) 스타벅스내에서의 머그컵

이와는 반대로 함수 외부에서 만든 변수는 프로그램이 살아있는 동안에 함께 살아있다가 프로그램이 종료되면 같이 소멸됩니다.
이렇게 프로그램 전체를 유효범위로 가지는 변수를 '전역변수'라고 합니다.

예 : 텀블러

예제1)

```
strdata = '전역변수' #func1 함수 외부에 있는 변수 (텀블러)

def func1( ):
    strdata2 = '지역변수' #func1 함수 내부에 있는 변수 : 스타벅스 머그컵
    return strdata2

print ( func1( ) ) #지역변수
```

*설명 : 대부분 많은 코드들이 지역변수를 사용합니다.
그러면 '전역변수'를 사용하는 경우는 언제 입니까?
---> 프로그램 전체에서 공통적으로 사용되고 잘 변하지 않는 데이터는 전역변수로 사용합니다.

예제:

```
    pi = 3.141592653589793

    def cycle_func1(num1): #원의 넓이를 구하는 함수
        global pi           #전역변수 pi를 함수내부로 가져올 수 있습니다.
                            # global이라는 키워드를 앞에 쓰면 됩니다.
        return pi * num1 * num1


    def cycle_fun2(num1): #원의 1/4인 부채꼴의 넓이를 구하는 함수
        global pi
        return 1/4 * pi * num1 * num1

    >>>
    print(cycle_func1(5))  #78.53981633974483
    print(cycle_fun2(5))   #19.634954084936208
```

■ 42. 함수 리턴값 이해하기(return)

모든 함수는 이름을 갖고 있습니다. 이 이름을 불러주면 파이썬은 그 이름 아래에 정이되어 있는 코드를 실행합니다. 이 때 함수를 부르는 코드를 호출자라고 합니다.
함수가 호출자에게 결과를 반환(return)이라고 합니다.

```
def cycle_func1(num1):
    global pi
    return pi * num1 * num1

print(cycle_func1(5))  #함수를 호출하는 호출자 코드 입니다.
```

***절대값을 출력하는 함수 (무조건 양수로 출력하는 함수)**

```
print(abs(-9)) #9
print(abs(9))  #9
```

■ 43. 파이썬 모듈 이해하기

파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 합니다.
이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게 하나의 파일로 묶어 놓은 것을 모듈이라고 합니다.
외부의 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로 가져와서 자유롭게 사용할 수 있도록 해야하는데 이런일을 파이썬에서는 모듈을 import한다 라고 합니다.

예제:

```
import time  # time모듈을 임폴트 합니다.
print('5초간 프로그램을 정지합니다')
time.sleep(5) # 5초동안 멈춰있는 것입니다.
print('5초가 지났습니다.')
```

예제 : 평균이 30이고 표준편차가 5인 가우시안 정규분포를 따르는 모집단을 백만개 구성하는 코드

```
import numpy as np  # numpy 모듈을 임폴트 합니다.

avg = 30
std = 5
N = 10000000
mogip = np.random.randn(N) * std + avg
print(mogip)
```

12/2 (044-051)

2020년 12월 2일 수요일 오전 9:46

모듈이란 ?
파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 합니다.
이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게
하나의 파일로 묶어 놓은것을 모듈이라고 합니다.
외부의 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로
가져와서 자유롭게 사용할 수 있도록 해야하는데 이런일을 파이썬에서는
모듈을 import 한다 라고 합니다.

■ 내가 직접 모듈 생성하기
지금 현재작성한 코드가 어느 디렉토리에 어느 이름으로 저장되었는지 확인하는 방법?

```
def add_number(n1, n2):
        result = n1 + n2
        return n1 +  n2

print(add_number(1,2))
```

c:₩₩users₩₩std₩₩.spyder-py3₩₩temp.py
↑
숨김폴더
--> . 이 있다는 것은 숨김폴더로 만들어졌다는 것이다.

```
def add_number(n1, n2):
        result = n1 + n2
        return n1 +  n2
```

위의 함수 스크립트를 c:₩₩Users₩₩std₩₩ 밑에 my_cal.py라는 이름으로 저장을 함

메뉴에 새로운 창 엽니다.
새로운 창에서 아래와 같이 임폴트를 합니다.

```
import my_cal                          #my_cal 모듈을 임폴트합니다.

print(my_cal.add_number(1,2))     #my_cal 모듈안에 add_number함수를 실행해라
```

결과 : 3

설명 : 이번 창에서는 def로 add_number 함수를 만드는 코드는 없습니다. my_cal.py 모듈안에 있는데 그 모듈을 이 창에서 사용할 수 있도록 import를 했습니다.

■ 44. 파이썬 패키지 이해하기

우리가 음악파일을 저장할 때도 장르별로 폴더를 만들어서 별도로 저장을 하듯이 파이썬 모듈도 음악처럼 개수가 많아지면 폴더(모듈 꾸러미)별로 별도로 관리를 해야 관리자가 편해지는데 이 폴더가 바로 '패키지'이다.

패키지(폴더) vs 모듈(폴더 안의 my_cal.py 같은 파이썬 스크립트)

□ 파이썬 패키지를 만드는 단계

1. 아래의 디렉토리에 my_loc라는 폴더를 생성한다.
   C:₩₩Users₩₩stu 혹은
   C:/Users/miseu/ 에

   c:₩₩Users₩₩stu₩my_loc라고 생성

2. my_loc 폴더 안에 my_cal.py를 옮겨 놓는다.
   (my_cal.py를 my_loc폴더에 복사하고 기존에 있던 my_cal.py는 삭제한다.)

3. 이 평범한 폴더가 패키지로 인정을 받으려면 반드시 갖고 있어야하는 파일이 있습니다. 그 파일이
   __init__.py라는 파일입니다.
   C:/Users/miseu/my_loc
   |
   1. __init__.py
   2. my_cal.py

4. 새로운 창에서 아래와 같이 스크립트를 수행합니다.

   ```
   from my_loc import my_cal       #from 패키지 import 모듈
   print(my_cal.add_number(1,2))  #my_cal 모듈 안에 있는 add_number함수를 실행해라~
   ```

   12/1일에 했던 것 중 비슷한 스크립트:
   from scipy.stats import norm #scipy 패키지 안에 stats라는 패키지의 norm이라는 모듈을 import해라~

   print(norm.pdf(x, 평균, 표준편차)) #norm모듈에 pdf(확률밀도함수)를 실행해라~

■ 45. 파이썬 모듈 임포트 이해하기 ① (import)

이미 만들어져 있는 어떤 함수를 우리가 작성하는 코드에서 자유롭게 활용할 수 있으려면 해당 함수가 포함된 모듈을 import 해야합니다. import하는 방법은 다음과 같습니다.

import 모듈이름
from 패키지 import 모듈이름
import 패키지이름.모듈이름

Ex) 아래와 같이 우리가 만든 모듈이 아니라 다른 사람이 만든 모듈을 import해서 썼는데 이 모듈은 어디에 있는 것일까?

```
import pandas
import numpy
```

위와 같이 패키지 이름을 안주고 모듈만 import했는데 잘 실행이 되었습니다.
위와같은 모듈은 어떤 모듈입니까?

1. 파이썬 내장모듈
2. sys.path에 정의되어 있는 모듈

- import를 만나면 파이썬 모듈을 찾는 순서

1. 파이썬 내장 모듈에 있는지 확인
2. sys.path에 정의되어 있는 디렉토리를 뒤져봅니다.

- 파이썬 내장 모듈이 무엇이 있는지 확인하는 방법

```
import sys
print( sys.builtin_module_names)
```

('_abc', '_ast', '_bisect', '_blake2', '_codecs', '_codecs_cn', '_codecs_hk', '_codecs_iso2022', '_codecs_jp', '_codecs_kr', '_codecs_tw', '_collections', '_contextvars', '_csv', '_datetime', '_functools', '_heapq', '_imp', '_io', '_json', '_locale', '_lsprof', '_md5', '_multibytecodec', '_opcode', '_operator', '_pickle', '_random', '_sha1', '_sha256', '_sha3', '_sha512', '_signal', '_sre', '_stat', '_statistics', '_string', '_struct', '_symtable', '_thread', '_tracemalloc', '_warnings', '_weakref', '_winapi', '_xxsubinterpreters', 'array', 'atexit', 'audioop', 'binascii', 'builtins', 'cmath', 'errno', 'faulthandler', 'gc', 'itertools', 'marshal', 'math', 'mmap', 'msvcrt', 'nt', 'parser', 'sys', 'time', 'winreg', 'xxsubtype', 'zlib')

- sys.path에 정의된 디렉토리가 무엇인지 확인하는 방법

```
import sys
for i in sys.path:
        print(i)
```

C:₩Users₩miseu₩anaconda3₩python38.zip

C:₩Users₩miseu₩anaconda3₩DLLs
C:₩Users₩miseu₩anaconda3₩lib
C:₩Users₩miseu₩anaconda3

C:₩Users₩miseu₩anaconda3₩lib₩site-packages
C:₩Users₩miseu₩anaconda3₩lib₩site-packages₩win32
C:₩Users₩miseu₩anaconda3₩lib₩site-packages₩win32₩lib
C:₩Users₩miseu₩anaconda3₩lib₩site-packages₩Pythonwin
C:₩Users₩miseu₩anaconda3₩lib₩site-packages₩IPython₩extensions
C:₩Users₩miseu₩.ipython

- site-packages란 무엇인가?
  site-packages란 파이썬의 기본 라이브러리 패키지 외에 추가적인 패키지를 설치하는 디렉토리 입니다.
  site-packages 디렉토리에 여러가지 소프트웨어 사용할 공통 모듈을 넣어두면 물리적인 장소에 구애받지 않고 모듈에 접근하여 반입할 수 있습니다

  아래의 명령아가 수행되려면 아래의 명령어를 수행하는 스크립트가 c:₩₩Users₩₩stu 밑에 있어야 합니다. 왜냐하면 my_loc폴더가 바로 c:₩₩Users₩₩stu 밑에 있기 때문입니다.
  from my_loc import my_cal

  그런데 c:₩₩Users₩₩stu가 아니더라도 다른 디렉토리에서라도 from my_loc import my_cal 명령어를 자유롭게 실행하려면 my_loc폴더가 C:₩Users₩miseu₩anaconda3₩Lib₩site-packages 밑에 있으면 됩니다.

■ 46. numpy 모듈 이해하기

- numpy 모듈이란? python언어에서 기본적으로 지원하지 안흔ㄴ 배열(array) 또는 행렬(matrix)의 계산을 쉽게 해주는 라이브러리 입니다.
  딥러닝에서 많이 사용하는 선형대 수학에 관련된 수식들을 python에서 쉽게 프로그래밍 할 수 있게 해줍니다.

Ex) 아래의 행렬을 만드시오!

1. 리스트로만 했을때

   ```
   a = [ [1,2], [4,7] ]
   print(a)
   ```

2. numpy array로 했을때

   ```
   import numpy as np
   a = [ [ 1,2], [4,7] ] #a리스트를 numpy배열로 구성함
   a2 = np.array(a)
   print(a2)
   ```

■ 47. numpy 모듈 사용하기2

numpy모듈로 최댓값, 최솟값, 평균값, 중앙값, 최빈값, 분산값, 표준편차, 공분산값, 상관계수의 통계값들을 쉽게 출력할 수 있습니다.

예제1. 아래의 리스트에서 최댓값을 출력하시오.

```
a = [28, 23, 21, 29, 30, 40, 23, 21]

import numpy as np
a2 = np.array(a)
print( np.max(a2))
```

예제2. 아래의 리스트에서 최댓값, 최솟값, 평균값, 중앙값을 출력하시오!

```
a = [28, 23, 21, 29, 30, 40, 23, 21]

import numpy as np
from scipy.stats import mode

a2 = np.array(a)
print( np.max(a2))
print( np.min(a2))
print( np.mean(a2))   #평균값
print( np.median(a2)) # 중앙값
print(mode(a)) # ModeResult(mode=array([21]), count=array([2]))
```

■ 48. 파일 열고 닫기(open, close)

파일은 텍스트 파일과 바이너리 파일 두가지 종류가 있습니다.
텍스트 파일은 사람이 읽을 수 잇는 글자로 저장된 파일이고 바이너리 파일은 컴퓨터가 읽고 이해할 수 있는 이진 데이터를 기록한 파일입니다.
예를들어 윈도우에서 제공하는 메모장 프로그램을 이용하여 내용을 저장하면 텍스트 파일로 저장됩니다.
이미지 뷰어로 볼 수 있는 jpg 이미지 파일은 이진 데이터를 jpg형식의 파일로 저장한 바이너리 파일입니다. 파이썬에 파일을 다루기 위해서 가장 먼저 해야할 일은 파일을 오픈 하는 것 입니다. 파일을 오픈하기 위해서는 open() 함수를 이용합니다.

문법 : open(파일이름, 모드)

모드	설명
r	텍스트 모드로 읽기
w	텍스트 모드로 쓰기
rb	바이너리 모드로 읽기
wb	바이너리 모드로 쓰기

예제1. 이미지 파일을 파이썬에서 여는 방법

1. lena.png파일을 내려받아 c:₩₩data₩₩ 밑에 저장합니다.
2. 아래의 코드를 실행 합니다.

```
import PIL.Image as pilimg #이미지를 파이썬에서 시각화 하기 위한 모듈
import numpy as np
import matplotlib,pyplot as plt # 데이터 시각화 전물 모듈을 임포트 합니다.

im = pilimg.open('c:₩₩data₩₩lena.png') #lena.png 파일을 읽어서 im에 입력
pix = np.array(im) #numpy 배열로 변환 합니다.
plt.imshow(pix)     # 화면에 띄웁니다.
```

■ 49. 클래스 이해하기(class)

객체 지향 프로그램에서 중요한 단어가 바로 클래스 입니다.
클래스는 프로그래머가 지정한 이름을 만든 하나의 독립된 공간이며, 이름공간(name space)이라 부릅니다.
클래스를 구성하는 주요 요소는 클래스에서 변수 역할을 하는 클래스 맴버와 함수와 동일한 역할을 하는 클래스 메소드 입니다.

클래스는 설계도이고 객체는 설계도를 바탕으로 만든 제품

예: 총 설계도	vs	총 (제품)
(클래스)		(객체)
구글 이미지		구글 이미지

예제1. 총 클래스(설계도)를 생성하시오.

총의 기능(함수로 생성) : 1. 총알을 장전하는 기능, 2. 총쏘는 기능

설명: 파이썬 클래스를 만들 때 self키워드는 필수입니다. 안쓰면 에러가 납니다.
self의 뜻은 자기자신 입니다. 이 총 설계도로 총을 만들고 그 총을 사용할 때 charge와 shoot기능을 사용할 것인데 이 때 이 총이 내꺼라는 의미가 바로 self입니다.

총 생산하는 명령어 --> gun1 = Gun( ) #준혁이
gun2 = Gun( ) #한결
gun3 = Gun( ) #세원

준혁이가 gun1사서 다음과 같이 충전했습니다.

```
gun1.charge(100)
gun2.shoot(10)
```

실제로 총 설계도에는 charge(self, num)이라고 해서 입력 매개변수가 2개인데 총을 사용할 때 충전할때는 gun1.charge(100) 이렇게 매개변수 하나에만 값을 입력했습니다.
그러면 self 입력매개변수에는 자동으로 gun1이 들어갑니다.

```
class Gun(): #클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.
    def charge(self,num): #총알을 충전하는 함수
        self.bullet = num

    def shoot(self,num): #총을 쏘는 함수
        for i in range(num):
            if self.bullet>0:
                print('탕')
                self.bullet -=1
            elif self.bullet ==0:
                print('총알이 없습니다')
                break
```

예제2. 위에서 만든 총설계도를 가지고 총을 한정 만드시오!

```
gun1      =   Gun()
 ↑              ↑
제품(객체)  클래스(총 설계도)
```

예제3. gun1이라는 제품에 총알 10발 충전 합니다.

```
gun1.charge(10)
 ↑         ↑
제품이름  총설계도안의 함수이름
```

예제4. 총을 쏴보시오~

```
gun.shoot(3)
```

```
탕
탕
탕
```

■ 51. 클래스 메소드(함수) 이해하기

card 클래스의 charge와 consume은 메소드(기능)입니다.

```
예 : a = [1,2,3,4]
     print(type(a) ) #<class 'list'>
```

설명 : 리스트가 클래스 였다고 출력이 됩니다. 리스트가 class라는 말은 리스트 클래스 안에도 분명히 메소드(기능)가 존재할 것 입니다.

card.charge(1000)
  ↑   ↑
객체   메소드(기능)

a = [1,2,3,4]
a.append(5)
↑   ↑
객체 메소드(기능)

- 리스트 객체의 유용한 메소드
  메소드 설명

1. append( ) : 리스트 맨 끝에 새로운 요소를 추가할 때 사용
2. count( ) : 리스트에서 특정 요소의 개수를 카운트할 때 사용
3. insert( ) : 리스트의 특정위치에 요소를 입력할 때 사용
4. remove( ) : 리스트이 특정 요소를 제거할 때 사용
5. sort( ) : 리스트의 요소를 순차적으로 정렬할 때 사용
6. reverse( ) : 리스트의 요소를 역순으로 정렬할 때 사용
7. index( ) : 리스트의 특정 요소의 위치를 출력할 때 사용

a = [1, 2, 3, 1, 2, 2, 2, 3, 4] #a라는 리스트 객체가 생성됨
print(a.count(2)) #a 리스트에 숫자 2가 몇개가 있는지 조회

12/3 (052-057)

2020년 12월 3일 목요일 오전 9:46

*클래스를 사용해서 파이썬 코딩을 해야하는 이유?

객체 지향 언어의 장점인 상속을 활용할 수 있기 때문입니다.
객체 지향 언어는 프로그램 할 때 너무 유용하기 때문에 객체를 만드는 것을 지향(선호)하는 프로그램 언어입니다.

신용카드 발급 및 사용 프로그램을 구축하고 싶다면? (회사)

팀장님 ---> 카드의 중요 기본 기능을 담는 설계도(클래스)를 생성
클래스 이름 : Card( )
↓
팀원1(나) 팀원2(동료)
(영화 할인 카드 클래스 생성) (주유 할인 카드 클래스 생성)

만약 내가 팀장님이 만든 카드의 중요기능(메소드)을 담은 클래스(Card)를 상속받기만 하면 나는 팀장님이 코딩한 카드의 주요기능을 코딩하지 않아도 되고 나는 단순히 영화 할인에만 집중해서 코드를 구현하면 된다.

★ 상속 코드 : class Movie_Card(Card):

면접 문제: 객체 지향언어의 장점이 무엇입니까?
상속입니다.

※ 파이썬 클래스에서 사용하는 키워드인 self가 무엇입니까?

```
class  Card( ): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.
    def __init__(self): # 설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
        self.cash = 0  # 만들때만 작동하고 그 다음엔 작동안함
        print ( '카드가 만들어졌습니다' , self.cash, '원이 충전 되었습니다')

    def  charge( self, num ): # 카드를 충전하는 함수
        self.cash = num
        print ( num, '원이 충전되었습니다')

이준혁_card = Card()
양건준_card = Card()
이준혁_card.charge(100000) : #준혁이가 준혁이 카드를 10만원 충전했습니다.
양건준_card.charge(5000000) : #건준이가 건준이 카드에 500만원 충전함
```

설명 : 여기서 5000000은 num변수에 들어가고 양건준_card 객체 이름은 self매개변수에 들어가서
양건준_card에 5000000 충전했다는 것을 나타내는 것입니다.

gun1 = Gun( )
↑ ↑
제품 설계도
(객체) (클래스)

다른 제품(객체)를 생성하고 싶다면?
gun2 = Gun( ) #생성하면 된다.
gun2.charge(100)
gun2.shoot(3)
↑ ↑
객체 메소드(기능)

■52. 클래스 생성자 이해하기

클래스의 인스턴스 객체가 생성될 때 자동적으로 호출되는 메소드가 클래스 생성자 입니다. 클래스 생성자는 __init__(self) 입니다.

카드의 기능?

0. 카드가 처음 만들어질때 0원으로 충전되게끔 하는 기능
1. 충전
2. 소비

예제: class Card():

```
    def __init__(self): #생성자 메소드
        self.cash = 0
        print( '카드가 발급되었습니다.', self.cash, '가 충전되어있습니다')

    kms_card = Card() #객체(제품)를 생성하는 명령어
```

■ 53. 클래스 소멸자 이해하기

객체가 사라질 때 자동으로 호출되는 함수를 소멸자(__del__)이라고 합니다.

예: 총 클래스로 kms_gun1 이라는 총으로 만들고 총을 사용하다가 이제 총을 폐기하고 싶으면 del 명령어로 총을 폐기하면 되는데 이때 자동으로 작동되는 메소드(함수)가 소멸자 함수 입니다.

예제: class Gun():

```
        def__init__(self):
            self.bullet = 0
            print('총 한정이 생성되었습니다')

            def __del__(self):
                print('총이 폐기되었습니다')


    kms_gun1 = Gun()
```

```
del kms_gun1
총이 폐기 되었습니다
```

■ 54. 클래스 상속 이해하기

클래스는 상속이라 특성을 가지고 있는 이름 공간입니다.
클래스에서 상속이란 어떤 클래스가 가지고 있는 멤버(변수)나 메소드(함수)를 상속받는 클래스가 모두 사용할 수 있도록 해주는 것입니다.
상속을 해주는 클래스를 부모 클래스 또는 슈퍼 클래스라고 하고 상속을 받는 클래스를 자식 클래스 또는 서브 클래스라고 합니다.
부모 클래스로부터 상속을 받아 자식 클래스를 정의하는 방법은 다음과 같습니다.

예제: class 자식클래스이름(부모클래스이름):

설명 : 자식 클래스는 부모 클래스에서 정의된 모든 멤버와 메소드를 그대로 상속 받습니다.

1. 팀장님이 카드의 주요기능을 담고 있는 Card() 클래스를 생성한다.

2. 팀원인 나는 영화할인 카드 클래스를 만들것인데 팀장님이 구현한 카드의 주요기능을 Card()클래스로 부터 다 상속받고 나는 팀장님이 구현한 코드는 구현하지 않습니다. 나는 그냥 영화할인에 집중해서 메소드를 만들면 됩니다.

예제 : 팀장님이 만든 카드 class(카드 설계도) 부모 클래스
   1. 카드가 발급되었을 때 0원이 충전되게 하는 기능 : __init__(self)
   2. 카드를 충전하는 기능 : charge(self, num):
   3. 카드를 쓰는 기능 : consume(self, num)

```
class Card(): #부모 클래스
    def __init__(self):
        self.cash = 0
        print ('카드가 발급되었습니다.')

    def charge(self,num):
        self.cash += num
        print(num, '원이 충전되었습니다.')

    def consume(self,num):
        if self.cash >= num:   #잔액이 소비하는 돈보다 커야지만 쓸 수 있게 함
            self.cash -= num
            print(num, '원이 사용되었습니다.')
        else:
            print('잔액이 부족합니다.')

        kms_card1 = Card()
        kms_card1.charge(10000)
```

```
kms_card1.consume(2000)
```

■ 54. 클래스 멤버와 인스턴스 멤버 이해하기

클래스에서 선언된 변수는 클래스 멤버(변수)와 인스턴스 멤버(변수)가 있습니다. 클래스 멤버는 클래스 메소드 바깥에서 선언되고 인스턴스 멤버는 클래스 메소드 안에서 self와 함께 선언되는 변수입니다.

예제 : 사원이 입사하면 입사한 사원에 대한 이메일을 자동으로 생성하고 이름을 출력하는 함수와 월급을 인상하는 함수를 담는 클래스를 생성

```
class  Employees :   #옆에 괄호( )를 따로 안 쓴 이유는 밑의 __init__ 함수의 입력 매개변수가 여러개여서 입니다.
    raise_amount = 1.1  # 클래스 변수 (클래서 맴버) (클래스 내부에는 있지만 메소드 바깥쪽에 있는 변수)
    def __init__(self, first, last, pay): # 클래서(설계도)를 사용해서 객제(제품)을 만들 때 바로 작동되는 함수
        self.first = first
        self.last  = last
        self.pay   = pay
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'
        self.raise_amount = 1.1  # 인스턴스 변수

    def  full_name(self): #사원의 전체이름을 출력하는 함수
        print('{} {}'.format(self.first, self.last)) #앞의 중괄호 2개에 각각 self.first와 self.last에 있는 값이 입력됩니다.

    def  apply_raise(self): # 월급을 인상하는 함수
        self.pay = int( self.pay * self.raise_amount)                    #5000000*1.1 = 5500000



emp_chulsu = Employees('chulsu', 'kim', 5000000)
print(emp_chulsu.pay)   # 5000000/ emp_chulsu = 객체이름, pay = 인스턴스 변수이름
emp_chulsu.apply_raise()
print(emp_chulsu.pay)  # 5500000

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
emp_chulsu2.raise_amount = 1.2 #클래스의 인스턴스 변수에 접근해서 인스턴스 변수를 변경을 하였다.
print(emp_chulsu2.pay)   # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay)  # 6000000
-----------------------------------------------------------------------
class  Employees: #옆에 괄호( )를 따로 안 쓴 이유는 밑의 __init__ 함수의 입력 매개변수가 여러개여서 입니다.
    raise_amount = 1.1  # 클래스 변수 (클래서 맴버) 클래스 변수에는 self가 없습니다.
    def __init__(self, first, last, pay): #emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
        self.first = first                #위와 같이 객체(제품)를 생성할 때 입력값이 3개가 입력되면서
        self.last  = last                 #객체(제품)가 만들어 집니다.
        self.pay   = pay
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'
```

```
        self.raise_amount = 1.1  # 인스턴스 변수

    def  full_name(self): #사원의 전체이름을 출력하는 함수
        print  '{} {}'.format(self.first, self.last)

    def  apply_raise(self): # 월급을 인상하는 함수
        self.pay = int( self.pay * Employees.raise_amount)  # 클래스 변수를 사용
                                                    ↑
```

아까는 여기가 self였는데 지금은 클래스 이름인 Employees가 있다. 이자리에 인스턴스 변수가 아니라 클래스 변수를 사용했습니다.

```
emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)

emp_chulsu2.raise_amount = 1.2   #인스턴스 변수는 1.2로 변경이 되었다. 그러나 클래스의 변수를 변경한 것은
                                 #아니다
print(emp_chulsu2.pay)   # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay)  # 5500000
```

클래스 내의 변수에는 2가지가 있는데

1. 클래스 변수 : 메소드 바깥의 변수
   객체 생성이후에 안의 값을 변경할 수 없다.

2. 인스턴스 변수: 메소드 안의 변수
   객체 생성이후에 안의 값을 변경할 수 있다.

앞으로는 클래스를 활용해서 코딩을 하는 습관을 들일 필요가 있습니다.

■ 55. 예외처리 이해하기 ① (try~except)

프로그램을 작성하다 보면 뜻하지 않은 오류가 발생하는 코드가 있을 수 있습니다. 프로그램이 실행되는 동안 오류가 발생하면 프로그램이 더 이상 진행될 수 없는 상태가 되는데 이를 예외상황이라고 합니다.
프로그램에 예외가 발생하더라도 프로그램을 중단시키지 않고 예외에 대한 적절한 처리를 하여 프로그램을 계속 진행시킬 수 있도록 하는 구문이 try ~ except 입니다.

```
예제 : try :
            문제가 없을 경우 실행할 코드
       except :
            문제가 생겼을 때 실행할 코드
```

예제1 : try ~ except 를 사용해서 예외처리를 하지 않았을 때의 코드

```
def my_divide():
    x = input('분자의 숫자를 입력하세요~')
```

```
        y = input('분모의 숫자를 입력하세요~')
        return int(x)/ int(y)

print(my_divide() )
분자의 숫자를 입력하세요~ 10
분모의 숫자를 입력하세요~ 2

>>>5.0

print(my_divide() )
분자의 숫자를 입력하세요~ 10
분모의 숫자를 입력하세요~ 0

>>>ZeroDivisionError: division by zero (에러가 남)

예제2 : try ~ except를 사용했을 때의 예제
def my_divide():
        try:
                x = input('분자의 숫자를 입력하세요~')
                y = input('분모의 숫자를 입력하세요~')
                return int(x)/ int(y)
        except:
                return '당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.'
print(my_divide() )
```

설명 : try와 except사이에 코드가 잘 실행이 된다면 except이후의 문장을 실행하지 않고 try와 except사이의 코드가 실행이 안된다면 except이후의 문자을 실행한다.

■ 56. 예외처리 이해하기 ② (try~except~else)

어떤 로직을 수행할 때 오류상황이 아닐 경우에만 어떤 작업을 수행하는 코드를 작성해야 할 때가 있다.
이런 경우에 try ~ except ~ else구문을 활용합니다.

```
예제 : try:
                실행할 코드 블록
        except:
                예외처리할 코드 블록
        else:
                except 절을 만나지않았을경우 실행하는 코드 블록

예제1: 186번 문제 코드 가져오기.
try:
    num = int(input('숫자를 입력하세요~'))
    print(num**2)

except:
    print('잘못된 값을 입력하셨습니다')
```

```
else:
    print('결과 출력에 성공했습니다')
```

설명 : try~except 사이의 코드에 에러가 안났다면 else: 이후의 문장을 실행합니다.

```
>>>
숫자를 입력하세요~4
16
결과 출력에 성공했습니다
```

■57. 예외처리 이해하기 ③ (try~except~finally)

오류 발생 유무와 상관없이 어떤 코드를 무조건 실행 시키려면 try ~ except ~ finally 구문을 활용합니다. 무조건 실행시키는 코드는 finally 부분에 작성하면 됩니다.

예 :

```
try:
    print('안녕하세요~')
except:
    print('예외가 발생했습니다')
finally:
    print('저는 무조건 실행됩니다')
```

12/4 (058-067)

2020년 12월 4일 금요일 오전 9:46

■ 어제 배웠던 내용중에 질문이 많았던 것을 복습

1. 클래서 변수와 인스턴스 변수
2. a = []의 위치

■ 58. 예외처리 이해하기 ④ (try~except Exception as e)

코드에서 예외가 발생하면 이에 대한 자세한 내용을 파악하는 것이 중요합니다. 파이썬은 발생 가능한 예외에 대해 exception 객체로 미리 정의해두고 있는데 이에 대한 내용은 다음 링크에서 확인할 수 있습니다.

http://docs.python.org/3/library/exceptions.html

설명 : 위의 사이트에 보면 여러가지 다양한 에러에 대해서 미리 에러처리를 할 수 있도록 정의해놓은 예외들을 확인할 수 있습니다.

예제:

```
try:
    x = int(input('분자의 숫자를 입력하시오'))
    y = int(input('분모의 숫자를 입력하시오'))
    print(x/y)
except:
    print('잘못된 값을 입력하셔서 나누기를 시도하셨습니다')
```

설명 : 위의 경우에는 분모값을 입력할 때 숫자 0을 입력했을 때와 문자 a를 입력했을 때 똑같이 '잘못된 값을 입력하셔서 나누기를 시도하셨습니다.'가 출력되게 했는데 좀 더 구체화해서 분모값으로 0을 입력하면 '0으로 나눌 수 없습니다'가 나오고 '분모값으로 a를 입력하면 잘못된 값을 입력하셨습니다' 가 나오게 하고 싶다면?

```
try:
    x = int(input('분자의 숫자를 입력하시오'))
    y = int(input('분모의 숫자를 입력하시오'))
    print(x/y)
except ZeroDivisionError: #대소문자 맞추기
    print('0으로 나눌 수 없습니다')
```

```
try:
    x = int(input('분자의 숫자를 입력하시오'))
    y = int(input('분모의 숫자를 입력하시오'))
    print(x/y)
```

```
except ZeroDivisionError: #대소문자 맞추기
    print('0으로 나눌 수 없습니다')
except:
    print('잘못된 값을 입력하셨습니다')
```

■ 59. 예외처리 이해하기 ⑤ (try~except 특정 예외)

파이썬 입장에서 봤을때는 오류가 아닌데 프로그래머가 '이건 오류이다.' 라고 일부러 프로그램이 안돌게 오류메세지를 출력하는 경우에 사용합니다.

계산하는 프로그램
프로그램 코드.....
프로그램 코드..... <--- 오류가 발생했으면 끝내버림 (뒤에 돌리지도 않음)
프로그램 코드.....
프로그램 코드.....

설명 : 금융권 프로그램에서는 금액이 안맞는 프로그램이 있다면 사고로 이어지게 되므로 금액이 안맞으면 프로그램을 종료해라 라고 예외처리를 할 수 있다.

문법 : 실행코드

```
if 어떤조건:
    raise Exception('예외가 발생했습니다.') #여기서 종료
else:
    실행코드
```

■ 60. 사용자 입력받기(input)

파이썬 내장 함수 input은 사용자가 키보드로 입력한 값을 문자열로 리턴합니다. input()의 사용자 입력을 돕기 위한 안내 문구나 힌트등을 표시하는 문자열이 됩니다.

```
예 :  name = input('사원이름을 입력하세요~')
      num = int(input('숫자값을 입력하세요~'))
```

■ 61. 자료형 확인하기(type)

파이썬의 자료형은 하나의 클래스 입니다.
파이썬은 숫자나 문자, 문자열, 리스트, 튜플, 사전, 함수 등을 각각의 하나의 클래스로 취급합니다.
코드를 작성하다가 변수 이름만 보고 이 자료가 어떤 자료형이닞 확인해야 하는 경우가 있습니다. 이때 파이썬 내장함수인 type()을 활용하면 자료형을 쉽게 확인할 수 있습니다.

예제:

```
numdata = 57
print(type(numdata)) #<class 'int'>

numdata2 = 57.2
print( type(numdata2) ) #<class 'float'>

strdata = '파이썬'
print(type(strdata)) #<class 'str'>

a = [1,2,3,4]
print(type(a)) #<class 'list'>
```

■ 62. 나눗셈에서 나머지만 구하기(%)

나누기 연산에서 나머지만 구하는 연산을 %로 구합니다.

```
예: print( 12 % 3 ) #0
    print( 12%5 )   #2
```

■ 63. 몫과 나머지 구하기(divmod)

예제:

```
a = 1113
b = 23

result1, result2 = divmod(1113, 23) #divmod 함수가 출력하는 값이 2개여서 앞에 result1,
result2 변수를
   ↑        ↑                          #두개를 썼음
  몫       나머지
print(result1, result2)

#값 : 48 9  <--값이 2개가 출력되고 있습니다.
       ↑ ↑
      몫, 나머지

result1 = divmod(1113, 23)
print(result1) #(49,9)

result1, result2 = divmod(1113, 23)
print(result1, result2)
```

■ 64. Pandas 를 이용한 데이터 검색

SQL, Pandas를 자유롭게 사용할 수 있어야 합니다.
SQL ---> Pandas, Pandas ---> SQL

예제 : emp[['ename', 'sal']][emp['ename']=='SCOTT']
↑ ↑
컬럼 조건

■ 65. Pandas 를 이용한 조인

예제 : 이름과 부서위치를 출력하시오!

```
SQL > select e.ename, d.loc
            from emp e, dept d
            where e.deptno = d.deptno;
```

```
Pandas>
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, on = 'deptno')
#설명 : emp 데이터 프레임과 dept 데이터 프레임을 merge를 써서 조인시키는데
on='deptno' 를 이용해서
#연결고리가 되는 컬럼을 지정해주면 됩니다
print(result)
#
    index  empno   ename        job  ...    comm deptno       dname       loc
0       1   7839    KING  PRESIDENT  ...     NaN      10  ACCOUNTING  NEW YORK
1       3   7782   CLARK    MANAGER  ...     NaN      10  ACCOUNTING  NEW YORK
2      14   7934  MILLER      CLERK  ...     NaN      10  ACCOUNTING  NEW YORK
3       2   7698   BLAKE    MANAGER  ...     NaN      30       SALES   CHICAGO
4       5   7654  MARTIN   SALESMAN  ...  1400.0      30       SALES   CHICAGO
5       6   7499   ALLEN   SALESMAN  ...   300.0      30       SALES   CHICAGO
6       7   7844  TURNER   SALESMAN  ...     0.0      30       SALES   CHICAGO
7       8   7900   JAMES      CLERK  ...     NaN      30       SALES   CHICAGO
8       9   7521    WARD   SALESMAN  ...   500.0      30       SALES   CHICAGO
9       4   7566   JONES    MANAGER  ...     NaN      20    RESEARCH    DALLAS
10     10   7902    FORD    ANALYST  ...     NaN      20    RESEARCH    DALLAS
11     11   7369   SMITH      CLERK  ...     NaN      20    RESEARCH    DALLAS
12     12   7788   SCOTT    ANALYST  ...     NaN      20    RESEARCH    DALLAS
13     13   7876   ADAMS      CLERK  ...     NaN      20    RESEARCH    DALLAS

print(result[['ename','loc']])
#
    ename       loc
0    KING  NEW YORK
1   CLARK  NEW YORK
```

```
2   MILLER   NEW YORK
3    BLAKE    CHICAGO
4   MARTIN    CHICAGO
5    ALLEN    CHICAGO
6   TURNER    CHICAGO
7    JAMES    CHICAGO
8     WARD    CHICAGO
9    JONES     DALLAS
10    FORD     DALLAS
11   SMITH     DALLAS
12   SCOTT     DALLAS
13   ADAMS     DALLAS
```

■ 66. Pandas 를 이용한 서브쿼리

예제 : JONES 보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오!

```
SQL >
select ename, sal
       from emp
       where sal > (select sal
                           from emp
                           where ename = 'JONES');
```

```
Pandas>
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

jsal = emp['sal'][emp['ename']=='JONES'].values[0] #JONES월급을 jsal변수에 담는다
print(emp[['ename', 'sal']][emp['sal']>jsal])
```

```
jsal type: <class 'pandas.core.series.Series'>
```

■ 67. Pandas 와 오라클 그룹함수의 비교

```
SQL> select max(sal)
             from emp;
```

```
Pandas>
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['sal'].max())
```

```
설명 : emp['sal'].max() --> 최대월급
       emp['sal'].min() --> 최소월급
       emp['sal'].sum() --> 토탈월급
       emp['sal'].var() --> 분산값
```

emp['sal'].std() --> 표준편차값

12/7 (068-074)

2020년 12월 7일 월요일 오전 9:46

■ 파이썬 수업 복습

1. 파이썬 자료형 5가지
2. 파이썬 연산자
3. if문
4. loop문
5. 문자열에서 특정 위치의 철자 검색, 슬라이싱 검색
6. 파이썬으로 함수 생성
7. 파이썬으로 클래스 생성
8. 파이썬의 예외처리
9. pandas 사용법

클래스를 왜 사용해야 하는지? (객체지향 언어의 장점이 무엇인지?)
답 : 상속을 사용할 수 있기 때문이다.

***상속을 사용했을 때 어떤 장점이 있습니까?**

답: 부모 클래스가 작성한 기능(메소드)을 코딩하지 않아도 됩니다. 중요한 기능은 부모 클래스를 작성하는 팀장님이 구현하고 팀원인 나는 팀장님이 구현한 중요기능을 그대로 상속받고 팀장님이 시킨 상세코드에 집중하면 되기 때문입니다.

***판다스로 조인하기 복습**

판다스로 조인(join)을 하기 위해서는 merge를 사용하면 됩니다.

문법 : pandas.merge(emp_left, dept_right, how = 'inner', on = 'deptno')
옵션 : how = 'inner' : emp와 dept 데이터 프레임에 공통적으로 존재하는 교집합일 경우에만 추출하겠다.
how = 'outer' : 열의 데이터가 양쪽 데이터 프레임에 공통적으로 존재하는 교집하이 아니어도 추출하겠다.
how = 'left' : 왼쪽 데이터 프레임의 키열에 속하는 데이터값을 기준으로 병합하겠다.
how = 'right' : 오른쪽 데이터 프레임의 키열에 속하는 데이터 값을 기준으로 병합하겠다.

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
result = pd.merge(emp, dept, on = 'deptno', how = 'left')
print(result[ ['ename', 'loc'] ])
```

=

```
select e.ename, d.loc
        from emp e, dept d
        where e.deptno = d.deptno(+)
```

둘이 같다.

■68 반올림수 구하기(round)

파이썬 내장함수 round( )는 인자로 입력된 숫자형 자리수에서 반올림한 결과를 리턴합니다.

예제 :

```
print( round(16.554) )    #소수점 첫번째 자리에서 반올림 17
print( round(16.554, 0) ) #소수점 첫번째 자리에서 반올림 17
print( round(16.554, 1) ) # 소수점 두번째 자리에서 반올림 16.6
print( round(16.554, 2) ) # 소수점 세번째 자리에서 반올림 16.55
```

*파이썬에서 반올림할 때 중요하게 알아야 할 내용

```
print( round(142.5) ) #143으로 예상했는데 142가 나온다.
"R과 파이썬은 짝수를 좋아합니다."
142.5 ------------> 142     2          2.5          3
187.5 ------------> 188     7          7.5          8
```

파이썬은 기본적으로 이것을 해결하는 함수가 없습니다.
0.5일때는 짝수를 좋아하게끔 반올림되고 0.51일때는 그냥 반올림 됩니다.

예제 : 판다스를 이용하지 않고 파이썬으로만 emp3.csv에서 이름과 월급을 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv") #os에 있는 emp2.csv를 읽어서 file이라는 변수에 넣는다
emp_csv = csv.reader(file) #file변수에 있는 csv파일을 읽어서 emp_csv 변수에 넣는다
print(emp_csv) #이상태에서 그냥 프린트하면 메모리 주소가 나옵니다.

for emp_list in emp_csv: #csv파일의 내용을 한행씩 리스트에 담아서
    print(emp_list)   #출력합니다
```

결과:

```
순서:     0        1        2        3      4          5      6        7
['', 'empno', 'ename', 'job', 'mgr', 'hiredate', 'sal', 'comm', 'deptno']
['0', '7839', 'KING', 'PRESIDENT', '', '1981-11-17', '5000', '', '10']
['1', '7698', 'BLAKE', 'MANAGER', '7839.0', '1981-05-01', '2850', '', '30']
['2', '7782', 'CLARK', 'MANAGER', '7839.0', '1981-05-09', '2450', '', '10']
['3', '7566', 'JONES', 'MANAGER', '7839.0', '1981-04-01', '2975', '', '20']
['4', '7654', 'MARTIN', 'SALESMAN', '7698.0', '1981-09-10', '1250', '1400.0', '30']
['5', '7499', 'ALLEN', 'SALESMAN', '7698.0', '1981-02-11', '1600', '300.0', '30']
['6', '7844', 'TURNER', 'SALESMAN', '7698.0', '1981-08-21', '1500', '0.0', '30']
['7', '7900', 'JAMES', 'CLERK', '7698.0', '1981-12-11', '950', '', '30']
['8', '7521', 'WARD', 'SALESMAN', '7698.0', '1981-02-23', '1250', '500.0', '30']
['9', '7902', 'FORD', 'ANALYST', '7566.0', '1981-12-11', '3000', '', '20']
['10', '7369', 'SMITH', 'CLERK', '7902.0', '1980-12-09', '800', '', '20']
['11', '7788', 'SCOTT', 'ANALYST', '7566.0', '1982-12-22', '3000', '', '20']
['12', '7876', 'ADAMS', 'CLERK', '7788.0', '1983-01-15', '1100', '', '20']
['13', '7934', 'MILLER', 'CLERK', '7782.0', '1982-01-11', '1300', '', '10']`
```

원하는 데이터를 얻으려면?

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
print(emp_csv)
for emp_list in emp_csv:
    print(emp_list[1], emp_list[5])
```

이렇게 해주어야 함

■ 69. 실수형 자료를 정수형 자료로 변환하기(int)

*파이썬의 변환함수 정리

	데이터 유형	변환함수
정수형	int	int()
문자형	str	str()
실수형	float	float()
리스트형	list	list()
튜플형	tuple	tuple()

*코드 작성시 수학연산을 하다보면 정수끼리만 계산해야하는 경우가 있습니다. 이 때 우리가 가진 데이터가 실수형이라면 실수형 자료를 정수형으로 변환한 후에 계산을 해줘야 합니다.
파이썬 내장함수 int()는 인자로 입력된 실수형 자료를 정수형 자료로 변환해줍니다. int()는 입력된 실수형 자료의 소수부분은 버리고 정수부분만 취하여 정수값으로 리턴합니다.

예제 : print(int(-5.4))

■ 70. 정수형 자료를 실수형 자료로 변환하기(float)

이미지 처리나 공학용 프로그램을 작성할 때 실수형끼리만 계산해야하는 경우가 많습니다. 이때 우리가 가진 데이터가 정수형이라면 정수형 자료를 실수형 자료로 변환한 후에 계산해 주어야 합니다.
파이썬 내장함수 float()은 인자로 입력된 정수형 자료를 실수형으로 변환해 줍니다.

예 : print(float(10)) #10.0

■ 71. 정수 리스트에서 소수만 걸러내기(filter)

파이썬 내장함수인 filter는 리스트와 같은 자료형에서 특정 조건을 만족하는 값만 편리하게 추출할 수 있는 방법을 제공합니다.
filter( )의 첫번째 인자는 특정 조건의 값을 추출하는 함수가 입력되며 두번째 인자(입력매개변수)는 리스트와 같은 자료형이 입력됩니다.

예제 : 숫자가 나열되어 있는 리스트에서 짝수만 추출해내는 코드를 작성

```
a = [1,2,3,4,5,6,7,8,9,10]
```

1. 숫자를 입력하면 짝수이면 결과를 출력하고 홀시면 출력하지 않는 함수 생성

```
def get_even(num):
    if num%2 == 0:
        return num
```

```
        else:
            return  #리턴 다음에 아무것도 쓰지 않아서 아무것도 리턴되지 않습니다.

print(get_even(2)) #2
print(get_even(7)) #None
```

2. filter함수와 get_even함수를 이용해서 위의 a리스트에서 짝수를 추출하기

```
a = [1,2,3,4,5,6,7,8,9,10]

result = filter(get_even, a)
                    ↑     ↑
               함수이름  리스트이름
print(result)
<filter object at 0x000001D9EDDC69A0>


a = [1,2,3,4,5,6,7,8,9,10]

result = filter(get_even, a)

print(list(result)) #list함수로 리스트로 변환 해야 결과가 출력됩니다. [2, 4, 6, 8, 10]
```

■ 72. 최대값, 최소값 구하기(max, min)

max( )와 min( )은 인자로 입력된 자료에서 최대, 최소값을 구해주는 함수 입니다. max( )와 min( )의 인자로 리스트와 같은 시퀀스 자료가 입력되면 시퀀스 자료의 요소 가운데 최대값 또는 최소값을 구하여 리턴합니다.

오라클	vs	파이썬
max		max
min		min
count		len
sum		sum
avg		mean

예제 :

```
a = [8, 7, 12, 55, 21, 34, 15, 9, 22]
print( max(a) )
```

■ 73. 판다스에서 결측치(NaN) 확인하기

emp3.csv에 comm처럼 결측치(NaN)를 처리하는 판다스 함수를 배웁니다.

↓

Not a Number

예 : 판다스를 이용해서 이름과 커미션을 출력하시오!

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[ ['ename', 'comm'] ])
```

■ 74. 판다스에서 파생변수 추가하는 방법

파생변수란 기존의 데이터를 가지고 새롭게 가공해서 만든 새로운 컬럼입니다.

예제 : 판다스에서 컬럼 추가하는 방법

emp 데이터 프레임에 sal의 데이터와 똑같은 데이터로 sal2라는 컬럼을 추가하려면?

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
emp['sal2'] = emp['sal']  # emp 데이터 프레임에 sal2 컬럼을 추가하는데 데이터는
                          # emp데이터 프레임의 sal로 하시오
print(emp)
```

12/8 (075-089)

2020년 12월 8일 화요일 오전 9:47

■ 75. 문자열에서 특정 위치의 문자 얻기

문자열에서 특정 위치의 문자를 얻는 방법을 인덱싱을 이용하는 것입니다. 인덱스는 0부터 시작합니다.
파이썬 인덱스는 음수도 가능합니다.

예제:

```
print('scott'[0]) #s
print('scott'[2]) #o

a = 'scott' #a라는 문자형 변수를 생성하면서 scott을 a라는 빈컵에 담았다.
print(a[0]) #s
print(a[2]) #o
```

■ 76. 문자열에서 지정한 구간의 문자열 얻기
문자열에서 특정 구간에 있는 문자열을 얻으려면 슬라이싱을 이용하면 됩니다.

예제 :

```
print('scott'[0:2]) #sc 0부터 2미만까지
a = 'scott'
print(a[0:2]) #sc
print(a[3:]) #tt 3부터 문자끝까지 출력됨
print(a[:3]) #sco 문자 처음부터 읽어서 3미만까지 출력
```

■77. 문자열에서 홀수 번째 문자만 추출하기

주어진 문자열에서 홀수번째 문자만 추출하는 방법은 슬라이싱의 스텝을 이용하면 됩니다.

예제:

```
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
result = txt[0:]  #0번째부터 문자끝까지 출력
print(result)
result2 = txt[0::2] #문자열에서 2칸씩 건너뛰면서 출력
print(result2)
```

■ 78. 문자열을 거꾸로 만들기

슬라이싱을 이용하면 매우 간단하게 거꾸로된 문자열을 얻을 수 있습니다. 문자열 txt를 처음부터 끝까지 스텝 -1로 슬라이싱 하면됩니다.

예제:
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
print(txt[:]) #문자열 전체 다 출력
print(txt[::]) #문자열 전체가 다 출력
print(txt[::1]) #문자열 전체가 다 출력되는데 처음부터 읽어서 끝까지 1스텝으로 읽는다.
result [: : -1] #문자열 전체를 뒤에 철자부터 앞으로 1스텝씩 읽는다~
print(result)

■ 79. 두 개의 문자열 합치기(+)

두개의 문자열을 합치는 방법은 매우 간단합니다.
문자열1 + 문자열2 + 문자열3으로 +연산자를 이용하면 됩니다.

예제:
a = 'scott'
b = 'king'
print(a+b) #scottking

■ 80. 문자열을 반복해서 새로운 문자열로 만들기(*)
예제:
print('★'*10) #★★★★★★★★★★
print('여러분~'*10)

■ 81. 문자열에서 특정 문자가 있는지 확인하기(in)

문자열에서 특정 문자가 있는지 없는지 확인하려면 in키워드를 이용합니다.

예제:
txt = 'abcdefghijklmnopqr'

if 'b' in txt:
    print('존재합니다.')
else:
    print('존재하지 않습니다')

■ 82. 문자열에서 특정 문자열이 있는지 확인하기(in)
보험회사에서 보험 상담원분들이 상담하실 때 상담내용중에 부적절한 용어가 있는지 없는지를 확인하여 금융감독원에 보고를 해야하는데 이 확인작업을 사람이 하려면 많은 인력과 시간이 드니까 파이썬으로 쉽게 할 수 있도록 프로그래밍할 때 문자열에서 문자열이 있는지 확인하는 코드를 작성하면 됩니다.

예제: 겨울왕국 스크립트에서 elsa라는 단어(문자열)가 몇번 나오는지 카운트 해봅니다. (파이썬 게시판에 winter.txt로 올려져 있습니다)

winter = open('c:₩₩data₩₩winter.txt')
print(winter)

결과:
<_io.TextIOWrapper name='c:₩₩data₩₩winter.txt' mode='r' encoding='cp949'>

예제1. 겨울왕국(winter.txt)스크립트를 파이썬으로 불러와서 출력하시오

```
winter = open('c:₩₩data₩₩winter.txt')
for i in winter:
    print(i.lower())
```

예제2. 출력되는 겨울왕국 스크립트를 전부 소문자로 출력하시오!

```
winter = open('c:₩₩data₩₩winter.txt')
for i in winter:      #winter.txt의 스크립트를 한행씩 가져오면서
    print(i.lower()) # 소문자로 변환해서 출력합니다.
```

예제3. 위에서 소문자로 출력된 스크립트 한행울 허너 가져와서 아래와 같이 실행하면 뭐가 나올까?

```
if 'elsa' in 'olaf skats and helps elsa coach anna.':
    print('존재합니다')
else:
    print('존재하지 않습니다.')
```

예제4. 겨울왕국 스크립트에 elsa가 몇번 나오는지 카운트 하시오!

```
winter = open('c:₩₩data₩₩winter.txt')
cnt = 0
for i in winter:      #winter.txt의 스크립트를 한행씩 가져오면서
    print(i.lower()) # 소문자로 변환해서 출력합니다.
-----------
winter = open('c:₩₩data₩₩winter.txt')
cnt = 0
for i in winter:
    if 'elsa' in i.lower():
        cnt = cnt+1
print(cnt)
```

>>318
이게 맞는 코드일까? No
한 행에 Elsa가 두번 나올 수도 있기 때문

```
Ex : if 'elsa' in 'olaf skates and helps elsa coach anna. elsa is princess':
        cnt = cnt+1
```

설명: 위의 코드는 한 행에 elsa가 두번 나와도 cnt는 1개만 카운트 됩니다. 정확하게 하려면 위의 문자열에 elsa가 2개 나오니까 cnt가 2개갸 되어야 합니다.

예제5. winter.txt를 한행씩 읽어서 어절 단위로 출력되게 하시오!

```
winter = open('c:₩₩data₩₩winter.txt')
winter2 = winter.read().split(' ') #스크립트를 공백단위로 분리해라~ split다음 괄호에 꼭 한칸 띄워줘야함
for i in winter2:
    print(i)
```

예제6. 위의 어절들이 소문자로 출력되게 하시오.

```
winter = open('c:₩₩data₩₩winter.txt')
winter2 = winter.read().split(' ')
for i in winter2:
    print(i.lower())
```

■ 83. 문자열 길이 구하기(len)

예제:

```
a = 'scott'
print( len(a) ) #5
```

■ 84. 문자열이 알파벳인지 검사하기(isalpha)

문자열은 문자나 숫자, 기호들로 구성이 됩니다. 코드를 작성하다 보면 특정 문자열이 한글이나 알파벳과 같이 사람의 언어를 표현하기 위해 사용되는 문자로만 구성되어 있는지 확인해야 하는 경우가 있습니다.
파이썬 문자열 객체가 제공하는 메소드인 isalpha()는 주어진 문자열이 사람의 언어 문자로만 구성되어 있는지 확인해줍니다.

예제:

```
txt1 = 'Warcarft three'
txt2 = '안녕'
txt3 = '3PO'

print(txt1.isalpha())  #False  (공백이 하나 있어서 False로 나옴)
print(txt2.isalpha())  #True   한글이나 다른나라 언어도 True로 나옴
print(txt3.isalpha())  #False  숫자3 때문에 False
```

■ 85. 문자열이 숫자인지 검사하기(isdigit)

문자열 객체의 isdigit() 매소드는 문자열을 구성하는 요소가 모두 숫자인지 체크하고 True 또는 False로 리턴합니다.

번호	내장함수	설명
1	isalpha()	알파벳(한글)이 맞는지 확인
2	isdigit()	숫자가 맞는지 확인
3	isspace()	공백이 맞는지 확인

■ 86. 문자열이 알파벳 또는 숫자인지 검사하기(isalnum)

알파벳(한글)과 숫자를 동시에 확인하는 문자열 함수는 isalnum 입니다.

예제:

```
a = 'A story is 2003'
for i in a:
```

```
        if i.isalnum() == True:
         print(i)
```

*케냐 은행의 데이터 분석 사례를 파이썬 코드로 구현해 봅니다.
(긍정단어집, 부정단어집을 다운로드 받으세요)

예제1. 긍정단어(positive-word.txt)를 파이썬으로 읽어서 한행씩 출력하시오.

```
positive = open("c:₩₩data₩₩positive-words.txt")
pos = positive.read().split('₩n')
for i in pos:
    print(i)
```

예제2. 아래의 pos를 프린트해보세요~

```
positive = open("c:₩₩data₩₩positive-words.txt")
pos = positive.read().split('₩n')  #positive 스크립트 행을 엔터로 구분한 단어들을.
print(pos)                         # pos에 담습니다.
```

pos의 타입: <class 'list'>

긍정단어들이 리스트에 담겼다. 리스트 이름은 pos입니다.

['', 'a+', 'abound', 'abounds', 'abundance', 'abundant', 'accessable', 'accessible', 'acclaim', 'acclaimed', 'acclamation', 'accolade', 'accolades', 'accommodative', 'accomodative', 'accomplish', 'accomplished', 'accomplishment', 'accomplishments', 'accurate', 'accurately', 'achievable', 'achievement', 'achievements', 'achievible', 'acumen', 'adaptable', 'adaptive', 'adequate', 'adjustable', 'admirable', 'admirably', 'admiration', 'admire', 'admirer', 'admiring', 'admiringly', 'adorable', 'adore', 'adored', 'adorer', 'adoring', 'adoringly', 'adroit', 'adroitly', 'adulate', 'adulation', 'adulatory', 'advanced', 'advantage', 'advantageous', 'advantageously', 'advantages', 'adventuresome', 'adventurous', 'advocate', 'advocated', 'advocates', 'affability', 'affable', 'affably', 'affectation', 'affection', 'affectionate', 'affinity', 'affirm', 'affirmation', 'affirmative', 'affluence', 'affluent', 'afford', 'affordable', 'affordably', 'afordable', 'agile', 'agilely', 'agility', 'agreeable', 'agreeableness', 'agreeably', 'all-around', 'alluring', 'alluringly', 'altruistic', 'altruistically', 'amaze', 'amazed', 'amazement', 'amazes', 'amazing', 'amazingly', 'ambitious', 'ambitiously', 'ameliorate', 'amenable', 'amenity', 'amiability', 'amiably', 'amiable', 'amicability', 'amicable', 'amicably', 'amity', 'ample', 'amply', 'amuse', 'amusing', 'amusingly', 'angel', 'angelic', 'apotheosis', 'appeal', 'appealing', 'applaud', 'appreciable', 'appreciate', 'appreciated', 'appreciates', 'appreciative', 'appreciatively', 'appropriate', 'approval', 'approve', 'ardent', 'ardently', 'ardor', 'articulate', 'aspiration', 'aspirations', 'aspire', 'assurance', 'assurances', 'assure', 'assuredly', 'assuring', 'astonish', 'astonished', 'astonishing', 'astonishingly', 'astonishment', 'astound', 'astounded', 'astounding', 'astoundingly', 'astutely', 'attentive', 'attraction', 'attractive', 'attractively', 'attune', 'audible', 'audibly', 'auspicious', 'authentic', 'authoritative', 'autonomous', 'available', 'aver', 'avid', 'avidly', 'award', 'awarded', 'awards', 'awe', 'awed', 'awesome', 'awesomely', 'awesomeness', 'awestruck', 'awsome', 'backbone', 'balanced', 'bargain', 'beauteous', 'beautiful', 'beautifullly', 'beautifully', 'beautify', 'beauty', 'beckon', 'beckoned', 'beckoning', 'beckons', 'believable', 'believeable', 'beloved', 'benefactor', 'beneficent', 'beneficial', 'beneficially', 'beneficiary', 'benefit', 'benefits', 'benevolence', 'benevolent', 'benifits', 'best', 'best-known', 'best-performing', 'best-selling', 'better', 'better-known', 'better-than-expected', 'beutifully', 'blameless', 'bless', 'blessing', 'bliss', 'blissful', 'blissfully', 'blithe', 'blockbuster', 'bloom', 'blossom', 'bolster', 'bonny', 'bonus', 'bonuses', 'boom', 'booming', 'boost', 'boundless', 'bountiful', 'brainiest', 'brainy', 'brand-new', 'brave', 'bravery', 'bravo', 'breakthrough', 'breakthroughs', 'breathlessness', 'breathtaking', 'breathtakingly', 'breeze', 'bright', 'brighten', 'brighter', 'brightest', 'brilliance', 'brilliances', 'brilliant', 'brilliantly', 'brisk', 'brotherly', 'bullish', 'buoyant', 'cajole', 'calm', 'calming', 'calmness', 'capability', 'capable', 'capably', 'captivate', 'captivating', 'carefree', 'cashback', 'cashbacks', 'catchy', 'celebrate', 'celebrated', 'celebration', 'celebratory', 'champ', 'champion', 'charisma', 'charismatic', 'charitable', 'charm', 'charming',

'charmingly', 'chaste', 'cheaper', 'cheapest', 'cheer', 'cheerful', 'cheery', 'cherish', 'cherished', 'cherub', 'chic', 'chivalrous', 'chivalry', 'civility', 'civilize', 'clarity', 'classic', 'classy', 'clean', 'cleaner', 'cleanest', 'cleanliness', 'cleanly', 'clear', 'clear-cut', 'cleared', 'clearer', 'clearly', 'clears', 'clever', 'cleverly', 'cohere', 'coherence', 'coherent', 'cohesive', 'colorful', 'comely', 'comfort', 'comfortable', 'comfortably', 'comforting', 'comfy', 'commend', 'commendable', 'commendably', 'commitment', 'commodious', 'compact', 'compactly', 'compassion', 'compassionate', 'compatible', 'competitive', 'complement', 'complementary', 'complemented', 'complements', 'compliant', 'compliment', 'complimentary', 'comprehensive', 'conciliate', 'conciliatory', 'concise', 'confidence', 'confident', 'congenial', 'congratulate', 'congratulation', 'congratulations', 'congratulatory', 'conscientious', 'considerate', 'consistent', 'consistently', 'constructive', 'consummate', 'contentment', 'continuity', 'contrasty', 'contribution', 'convenience', 'convenient', 'conveniently', 'convience', 'convienient', 'convient', 'convincing', 'convincingly', 'cool', 'coolest', 'cooperative', 'cooperatively', 'cornerstone', 'correct', 'correctly', 'cost-effective', 'cost-saving', 'counter-attack', 'counter-attacks', 'courage', 'courageous', 'courageously', 'courageousness', 'courteous', 'courtly', 'covenant', 'cozy', 'creative', 'credence', 'credible', 'crisp', 'crisper', 'cure', 'cure-all', 'cushy', 'cute', 'cuteness', 'danke', 'danken', 'daring', 'daringly', 'darling', 'dashing', 'dauntless', 'dawn', 'dazzle', 'dazzled', 'dazzling', 'dead-cheap', 'dead-on', 'decency', 'decent', 'decisive', 'decisiveness', 'dedicated', 'defeat', 'defeated', 'defeating', 'defeats', 'defender', 'deference', 'deft', 'deginified', 'delectable', 'delicacy', 'delicate', 'delicious', 'delight', 'delighted', 'delightful', 'delightfully', 'delightfulness', 'dependable', 'dependably', 'deservedly', 'deserving', 'desirable', 'desiring', 'desirous', 'destiny', 'detachable', 'devout', 'dexterous', 'dexterously', 'dextrous', 'dignified', 'dignify', 'dignity', 'diligence', 'diligent', 'diligently', 'diplomatic', 'dirt-cheap', 'distinction', 'distinctive', 'distinguished', 'diversified', 'divine', 'divinely', 'dominate', 'dominated', 'dominates', 'dote', 'dotingly', 'doubtless', 'dreamland', 'dumbfounded', 'dumbfounding', 'dummy-proof', 'durable', 'dynamic', 'eager', 'eagerly', 'eagerness', 'earnest', 'earnestly', 'earnestness', 'ease', 'eased', 'eases', 'easier', 'easiest', 'easiness', 'easing', 'easy', 'easy-to-use', 'easygoing', 'ebullience', 'ebullient', 'ebulliently', 'ecenomical', 'economical', 'ecstasies', 'ecstasy', 'ecstatic', 'ecstatically', 'edify', 'educated', 'effective', 'effectively', 'effectiveness', 'effectual', 'efficacious', 'efficient', 'efficiently', 'effortless', 'effortlessly', 'effusion', 'effusive', 'effusively', 'effusiveness', 'elan', 'elate', 'elated', 'elatedly', 'elation', 'electrify', 'elegance', 'elegant', 'elegantly', 'elevate', 'elite', 'eloquence', 'eloquent', 'eloquently', 'embolden', 'eminence', 'eminent', 'empathize', 'empathy', 'empower', 'empowerment', 'enchant', 'enchanted', 'enchanting', 'enchantingly', 'encourage', 'encouragement', 'encouraging', 'encouragingly', 'endear', 'endearing', 'endorse', 'endorsed', 'endorsement', 'endorses', 'endorsing', 'energetic', 'energize', 'energy-efficient', 'energy-saving', 'engaging', 'engrossing', 'enhance', 'enhanced', 'enhancement', 'enhances', 'enjoy', 'enjoyable', 'enjoyably', 'enjoyed', 'enjoying', 'enjoyment', 'enjoys', 'enlighten', 'enlightenment', 'enliven', 'ennoble', 'enough', 'enrapt', 'enrapture', 'enraptured', 'enrich', 'enrichment', 'enterprising', 'entertain', 'entertaining', 'entertains', 'enthral', 'enthrall', 'enthralled', 'enthuse', 'enthusiasm', 'enthusiast', 'enthusiastic', 'enthusiastically', 'entice', 'enticed', 'enticing', 'enticingly', 'entranced', 'entrancing', 'entrust', 'enviable', 'enviably', 'envious', 'enviously', 'enviousness', 'envy', 'equitable', 'ergonomical', 'err-free', 'erudite', 'ethical', 'eulogize', 'euphoria', 'euphoric', 'euphorically', 'evaluative', 'evenly', 'eventful', 'everlasting', 'evocative', 'exalt', 'exaltation', 'exalted', 'exaltedly', 'exalting', 'exaltingly', 'examplar', 'examplary', 'excallent', 'exceed', 'exceeded', 'exceeding', 'exceedingly', 'exceeds', 'excel', 'exceled', 'excelent', 'excellant', 'excelled', 'excellence', 'excellency', 'excellent', 'excellently', 'excels', 'exceptional', 'exceptionally', 'excite', 'excited', 'excitedly', 'excitedness', 'excitement', 'excites', 'exciting', 'excitingly', 'exellent', 'exemplar', 'exemplary', 'exhilarate', 'exhilarating', 'exhilaratingly', 'exhilaration', 'exonerate', 'expansive', 'expeditiously', 'expertly', 'exquisite', 'exquisitely', 'extol', 'extoll', 'extraordinarily', 'extraordinary', 'exuberance', 'exuberant', 'exuberantly', 'exult', 'exultant', 'exultation', 'exultingly', 'eye-catch', 'eye-catching', 'eyecatch', 'eyecatching', 'fabulous', 'fabulously', 'facilitate', 'fair', 'fairly', 'fairness', 'faith', 'faithful', 'faithfully', 'faithfulness', 'fame', 'famed', 'famous', 'famously', 'fancier', 'fancinating', 'fancy', 'fanfare', 'fans', 'fantastic', 'fantastically', 'fascinate', 'fascinating', 'fascinatingly', 'fascination', 'fashionable', 'fashionably', 'fast', 'fast-growing', 'fast-paced', 'faster', 'fastest', 'fastest-growing', 'faultless', 'fav', 'fave', 'favor', 'favorable', 'favored', 'favorite', 'favorited', 'favour', 'fearless', 'fearlessly', 'feasible', 'feasibly', 'feat', 'feature-rich', 'fecilitous', 'feisty', 'felicitate', 'felicitous', 'felicity', 'fertile', 'fervent', 'fervently', 'fervid', 'fervidly', 'fervor', 'festive', 'fidelity', 'fiery', 'fine', 'fine-looking', 'finely', 'finer', 'finest', 'firmer', 'first-class', 'first-in-class', 'first-rate', 'flashy', 'flatter', 'flattering', 'flatteringly', 'flawless', 'flawlessly', 'flexibility', 'flexible', 'flourish', 'flourishing', 'fluent', 'flutter',

'fond', 'fondly', 'fondness', 'foolproof', 'foremost', 'foresight', 'formidable', 'fortitude', 'fortuitous', 'fortuitously', 'fortunate', 'fortunately', 'fortune', 'fragrant', 'free', 'freed', 'freedom', 'freedoms', 'fresh', 'fresher', 'freshest', 'friendliness', 'friendly', 'frolic', 'frugal', 'fruitful', 'ftw', 'fulfillment', 'fun', 'futurestic', 'futuristic', 'gaiety', 'gaily', 'gain', 'gained', 'gainful', 'gainfully', 'gaining', 'gains', 'gallant', 'gallantly', 'galore', 'geekier', 'geeky', 'gem', 'gems', 'generosity', 'generous', 'generously', 'genial', 'genius', 'gentle', 'gentlest', 'genuine', 'gifted', 'glad', 'gladden', 'gladly', 'gladness', 'glamorous', 'glee', 'gleeful', 'gleefully', 'glimmer', 'glimmering', 'glisten', 'glistening', 'glitter', 'glitz', 'glorify', 'glorious', 'gloriously', 'glory', 'glow', 'glowing', 'glowingly', 'god-given', 'god-send', 'godlike', 'godsend', 'gold', 'golden', 'good', 'goodly', 'goodness', 'goodwill', 'goood', 'gooood', 'gorgeous', 'gorgeously', 'grace', 'graceful', 'gracefully', 'gracious', 'graciously', 'graciousness', 'grand', 'grandeur', 'grateful', 'gratefully', 'gratification', 'gratified', 'gratifies', 'gratify', 'gratifying', 'gratifyingly', 'gratitude', 'great', 'greatest', 'greatness', 'grin', 'groundbreaking', 'guarantee', 'guidance', 'guiltless', 'gumption', 'gush', 'gusto', 'gutsy', 'hail', 'halcyon', 'hale', 'hallmark', 'hallmarks', 'hallowed', 'handier', 'handily', 'hands-down', 'handsome', 'handsomely', 'handy', 'happier', 'happily', 'happiness', 'happy', 'hard-working', 'hardier', 'hardy', 'harmless', 'harmonious', 'harmoniously', 'harmonize', 'harmony', 'headway', 'heal', 'healthful', 'healthy', 'hearten', 'heartening', 'heartfelt', 'heartily', 'heartwarming', 'heaven', 'heavenly', 'helped', 'helpful', 'helping', 'hero', 'heroic', 'heroically', 'heroine', 'heroize', 'heros', 'high-quality', 'high-spirited', 'hilarious', 'holy', 'homage', 'honest', 'honesty', 'honor', 'honorable', 'honored', 'honoring', 'hooray', 'hopeful', 'hospitable', 'hot', 'hotcake', 'hotcakes', 'hottest', 'hug', 'humane', 'humble', 'humility', 'humor', 'humorous', 'humorously', 'humour', 'humourous', 'ideal', 'idealize', 'ideally', 'idol', 'idolize', 'idolized', 'idyllic', 'illuminate', 'illuminati', 'illuminating', 'illumine', 'illustrious', 'ilu', 'imaculate', 'imaginative', 'immaculate', 'immaculately', 'immense', 'impartial', 'impartiality', 'impartially', 'impassioned', 'impeccable', 'impeccably', 'important', 'impress', 'impressed', 'impresses', 'impressive', 'impressively', 'impressiveness', 'improve', 'improved', 'improvement', 'improvements', 'improves', 'improving', 'incredible', 'incredibly', 'indebted', 'individualized', 'indulgence', 'indulgent', 'industrious', 'inestimable', 'inestimably', 'inexpensive', 'infallibility', 'infallible', 'infallibly', 'influential', 'ingenious', 'ingeniously', 'ingenuity', 'ingenuous', 'ingenuously', 'innocuous', 'innovation', 'innovative', 'inpressed', 'insightful', 'insightfully', 'inspiration', 'inspirational', 'inspire', 'inspiring', 'instantly', 'instructive', 'instrumental', 'integral', 'integrated', 'intelligence', 'intelligent', 'intelligible', 'interesting', 'interests', 'intimacy', 'intimate', 'intricate', 'intrigue', 'intriguing', 'intriguingly', 'intuitive', 'invaluable', 'invaluablely', 'inventive', 'invigorate', 'invigorating', 'invincibility', 'invincible', 'inviolable', 'inviolate', 'invulnerable', 'irreplaceable', 'irreproachable', 'irresistible', 'irresistibly', 'issue-free', 'jaw-droping', 'jaw-dropping', 'jollify', 'jolly', 'jovial', 'joy', 'joyful', 'joyfully', 'joyous', 'joyously', 'jubilant', 'jubilantly', 'jubilate', 'jubilation', 'jubiliant', 'judicious', 'justly', 'keen', 'keenly', 'keenness', 'kid-friendly', 'kindliness', 'kindly', 'kindness', 'knowledgeable', 'kudos', 'large-capacity', 'laud', 'laudable', 'laudably', 'lavish', 'lavishly', 'law-abiding', 'lawful', 'lawfully', 'lead', 'leading', 'leads', 'lean', 'led', 'legendary', 'leverage', 'levity', 'liberate', 'liberation', 'liberty', 'lifesaver', 'light-hearted', 'lighter', 'likable', 'like', 'liked', 'likes', 'liking', 'lionhearted', 'lively', 'logical', 'long-lasting', 'lovable', 'lovably', 'love', 'loved', 'loveliness', 'lovely', 'lover', 'loves', 'loving', 'low-cost', 'low-price', 'low-priced', 'low-risk', 'lower-priced', 'loyal', 'loyalty', 'lucid', 'lucidly', 'luck', 'luckier', 'luckiest', 'luckiness', 'lucky', 'lucrative', 'luminous', 'lush', 'luster', 'lustrous', 'luxuriant', 'luxuriate', 'luxurious', 'luxuriously', 'luxury', 'lyrical', 'magic', 'magical', 'magnanimous', 'magnanimously', 'magnificence', 'magnificent', 'magnificently', 'majestic', 'majesty', 'manageable', 'maneuverable', 'marvel', 'marveled', 'marvelled', 'marvellous', 'marvelous', 'marvelously', 'marvelousness', 'marvels', 'master', 'masterful', 'masterfully', 'masterpiece', 'masterpieces', 'masters', 'mastery', 'matchless', 'mature', 'maturely', 'maturity', 'meaningful', 'memorable', 'merciful', 'mercifully', 'mercy', 'merit', 'meritorious', 'merrily', 'merriment', 'merriness', 'merry', 'mesmerize', 'mesmerized', 'mesmerizes', 'mesmerizing', 'mesmerizingly', 'meticulous', 'meticulously', 'mightily', 'mighty', 'mind-blowing', 'miracle', 'miracles', 'miraculous', 'miraculously', 'miraculousness', 'modern', 'modest', 'modesty', 'momentous', 'monumental', 'monumentally', 'morality', 'motivated', 'multi-purpose', 'navigable', 'neat', 'neatest', 'neatly', 'nice', 'nicely', 'nicer', 'nicest', 'nifty', 'nimble', 'noble', 'nobly', 'noiseless', 'non-violence', 'non-violent', 'notably', 'noteworthy', 'nourish', 'nourishing', 'nourishment', 'novelty', 'nurturing', 'oasis', 'obsession', 'obsessions', 'obtainable', 'openly', 'openness', 'optimal', 'optimism', 'optimistic', 'opulent', 'orderly', 'originality', 'outdo', 'outdone', 'outperform', 'outperformed', 'outperforming', 'outperforms', 'outshine', 'outshone', 'outsmart', 'outstanding', 'outstandingly', 'outstrip', 'outwit', 'ovation', 'overjoyed', 'overtake', 'overtaken', 'overtakes', 'overtaking', 'overtook', 'overture', 'pain-

free', 'painless', 'painlessly', 'palatial', 'pamper', 'pampered', 'pamperedly', 'pamperedness', 'pampers', 'panoramic', 'paradise', 'paramount', 'pardon', 'passion', 'passionate', 'passionately', 'patience', 'patient', 'patiently', 'patriot', 'patriotic', 'peace', 'peaceable', 'peaceful', 'peacefully', 'peacekeepers', 'peach', 'peerless', 'pep', 'pepped', 'pepping', 'peppy', 'peps', 'perfect', 'perfection', 'perfectly', 'permissible', 'perseverance', 'persevere', 'personages', 'personalized', 'phenomenal', 'phenomenally', 'picturesque', 'piety', 'pinnacle', 'playful', 'playfully', 'pleasant', 'pleasantly', 'pleased', 'pleases', 'pleasing', 'pleasingly', 'pleasurable', 'pleasurably', 'pleasure', 'plentiful', 'pluses', 'plush', 'plusses', 'poetic', 'poeticize', 'poignant', 'poise', 'poised', 'polished', 'polite', 'politeness', 'popular', 'portable', 'posh', 'positive', 'positively', 'positives', 'powerful', 'powerfully', 'praise', 'praiseworthy', 'praising', 'pre-eminent', 'precious', 'precise', 'precisely', 'preeminent', 'prefer', 'preferable', 'preferably', 'prefered', 'preferes', 'preferring', 'prefers', 'premier', 'prestige', 'prestigious', 'prettily', 'pretty', 'priceless', 'pride', 'principled', 'privilege', 'privileged', 'prize', 'proactive', 'problem-free', 'problem-solver', 'prodigious', 'prodigiously', 'prodigy', 'productive', 'productively', 'proficient', 'proficiently', 'profound', 'profoundly', 'profuse', 'profusion', 'progress', 'progressive', 'prolific', 'prominence', 'prominent', 'promise', 'promised', 'promises', 'promising', 'promoter', 'prompt', 'promptly', 'proper', 'properly', 'propitious', 'propitiously', 'pros', 'prosper', 'prosperity', 'prosperous', 'prospros', 'protect', 'protection', 'protective', 'proud', 'proven', 'proves', 'providence', 'proving', 'prowess', 'prudence', 'prudent', 'prudently', 'punctual', 'pure', 'purify', 'purposeful', 'quaint', 'qualified', 'qualify', 'quicker', 'quiet', 'quieter', 'radiance', 'radiant', 'rapid', 'rapport', 'rapt', 'rapture', 'raptureous', 'raptureously', 'rapturous', 'rapturously', 'rational', 'razor-sharp', 'reachable', 'readable', 'readily', 'ready', 'reaffirm', 'reaffirmation', 'realistic', 'realizable', 'reasonable', 'reasonably', 'reasoned', 'reassurance', 'reassure', 'receptive', 'reclaim', 'recomend', 'recommend', 'recommendation', 'recommendations', 'recommended', 'reconcile', 'reconciliation', 'record-setting', 'recover', 'recovery', 'rectification', 'rectify', 'rectifying', 'redeem', 'redeeming', 'redemption', 'refine', 'refined', 'refinement', 'reform', 'reformed', 'reforming', 'reforms', 'refresh', 'refreshed', 'refreshing', 'refund', 'refunded', 'regal', 'regally', 'regard', 'rejoice', 'rejoicing', 'rejoicingly', 'rejuvenate', 'rejuvenated', 'rejuvenating', 'relaxed', 'relent', 'reliable', 'reliably', 'relief', 'relish', 'remarkable', 'remarkably', 'remedy', 'remission', 'remunerate', 'renaissance', 'renewed', 'renown', 'renowned', 'replaceable', 'reputable', 'reputation', 'resilient', 'resolute', 'resound', 'resounding', 'resourceful', 'resourcefulness', 'respect', 'respectable', 'respectful', 'respectfully', 'respite', 'resplendent', 'responsibly', 'responsive', 'restful', 'restored', 'restructure', 'restructured', 'restructuring', 'retractable', 'revel', 'revelation', 'revere', 'reverence', 'reverent', 'reverently', 'revitalize', 'revival', 'revive', 'revives', 'revolutionary', 'revolutionize', 'revolutionized', 'revolutionizes', 'reward', 'rewarding', 'rewardingly', 'rich', 'richer', 'richly', 'richness', 'right', 'righten', 'righteous', 'righteously', 'righteousness', 'rightful', 'rightfully', 'rightly', 'rightness', 'risk-free', 'robust', 'rock-star', 'rock-stars', 'rockstar', 'rockstars', 'romantic', 'romantically', 'romanticize', 'roomier', 'roomy', 'rosy', 'safe', 'safely', 'sagacity', 'sagely', 'saint', 'saintliness', 'saintly', 'salutary', 'salute', 'sane', 'satisfactorily', 'satisfactory', 'satisfied', 'satisfies', 'satisfy', 'satisfying', 'satisified', 'saver', 'savings', 'savior', 'savvy', 'scenic', 'seamless', 'seasoned', 'secure', 'securely', 'selective', 'self-determination', 'self-respect', 'self-satisfaction', 'self-sufficiency', 'self-sufficient', 'sensation', 'sensational', 'sensationally', 'sensations', 'sensible', 'sensibly', 'sensitive', 'serene', 'serenity', 'sexy', 'sharp', 'sharper', 'sharpest', 'shimmering', 'shimmeringly', 'shine', 'shiny', 'significant', 'silent', 'simpler', 'simplest', 'simplified', 'simplifies', 'simplify', 'simplifying', 'sincere', 'sincerely', 'sincerity', 'skill', 'skilled', 'skillful', 'skillfully', 'slammin', 'sleek', 'slick', 'smart', 'smarter', 'smartest', 'smartly', 'smile', 'smiles', 'smiling', 'smilingly', 'smitten', 'smooth', 'smoother', 'smoothes', 'smoothest', 'smoothly', 'snappy', 'snazzy', 'sociable', 'soft', 'softer', 'solace', 'solicitous', 'solicitously', 'solid', 'solidarity', 'soothe', 'soothingly', 'sophisticated', 'soulful', 'soundly', 'soundness', 'spacious', 'sparkle', 'sparkling', 'spectacular', 'spectacularly', 'speedily', 'speedy', 'spellbind', 'spellbinding', 'spellbindingly', 'spellbound', 'spirited', 'spiritual', 'splendid', 'splendidly', 'splendor', 'spontaneous', 'sporty', 'spotless', 'sprightly', 'stability', 'stabilize', 'stable', 'stainless', 'standout', 'state-of-the-art', 'stately', 'statuesque', 'staunch', 'staunchly', 'staunchness', 'steadfast', 'steadfastly', 'steadfastness', 'steadiest', 'steadiness', 'steady', 'stellar', 'stellarly', 'stimulate', 'stimulates', 'stimulating', 'stimulative', 'stirringly', 'straighten', 'straightforward', 'streamlined', 'striking', 'strikingly', 'striving', 'strong', 'stronger', 'strongest', 'stunned', 'stunning', 'stunningly', 'stupendous', 'stupendously', 'sturdier', 'sturdy', 'stylish', 'stylishly', 'stylized', 'suave', 'suavely', 'sublime', 'subsidize', 'subsidized', 'subsidizes', 'subsidizing', 'substantive', 'succeed', 'succeeded', 'succeeding', 'succeeds', 'succes', 'success', 'successes', 'successful', 'successfully', 'suffice', 'sufficed', 'suffices', 'sufficient', 'sufficiently',

'suitable', 'sumptuous', 'sumptuously', 'sumptuousness', 'super', 'superb', 'superbly', 'superior', 'superiority', 'supple', 'support', 'supported', 'supporter', 'supporting', 'supportive', 'supports', 'supremacy', 'supreme', 'supremely', 'supurb', 'supurbly', 'surmount', 'surpass', 'surreal', 'survival', 'survivor', 'sustainability', 'sustainable', 'swank', 'swankier', 'swankiest', 'swanky', 'sweeping', 'sweet', 'sweeten', 'sweetheart', 'sweetly', 'sweetness', 'swift', 'swiftness', 'talent', 'talented', 'talents', 'tantalize', 'tantalizing', 'tantalizingly', 'tempt', 'tempting', 'temptingly', 'tenacious', 'tenaciously', 'tenacity', 'tender', 'tenderly', 'terrific', 'terrifically', 'thank', 'thankful', 'thinner', 'thoughtful', 'thoughtfully', 'thoughtfulness', 'thrift', 'thrifty', 'thrill', 'thrilled', 'thrilling', 'thrillingly', 'thrills', 'thrive', 'thriving', 'thumb-up', 'thumbs-up', 'tickle', 'tidy', 'time-honored', 'timely', 'tingle', 'titillate', 'titillating', 'titillatingly', 'togetherness', 'tolerable', 'toll-free', 'top', 'top-notch', 'top-quality', 'topnotch', 'tops', 'tough', 'tougher', 'toughest', 'traction', 'tranquil', 'tranquility', 'transparent', 'treasure', 'tremendously', 'trendy', 'triumph', 'triumphal', 'triumphant', 'triumphantly', 'trivially', 'trophy', 'trouble-free', 'trump', 'trumpet', 'trust', 'trusted', 'trusting', 'trustingly', 'trustworthiness', 'trustworthy', 'trusty', 'truthful', 'truthfully', 'truthfulness', 'twinkly', 'ultra-crisp', 'unabashed', 'unabashedly', 'unaffected', 'unassailable', 'unbeatable', 'unbiased', 'unbound', 'uncomplicated', 'unconditional', 'undamaged', 'undaunted', 'understandable', 'undisputable', 'undisputably', 'undisputed', 'unencumbered', 'unequivocal', 'unequivocally', 'unfazed', 'unfettered', 'unforgettable', 'unity', 'unlimited', 'unmatched', 'unparalleled', 'unquestionable', 'unquestionably', 'unreal', 'unrestricted', 'unrivaled', 'unselfish', 'unwavering', 'upbeat', 'upgradable', 'upgradeable', 'upgraded', 'upheld', 'uphold', 'uplift', 'uplifting', 'upliftingly', 'upliftment', 'upscale', 'usable', 'useable', 'useful', 'user-friendly', 'user-replaceable', 'valiant', 'valiantly', 'valor', 'valuable', 'variety', 'venerate', 'verifiable', 'veritable', 'versatile', 'versatility', 'vibrant', 'vibrantly', 'victorious', 'victory', 'viewable', 'vigilance', 'vigilant', 'virtue', 'virtuous', 'virtuously', 'visionary', 'vivacious', 'vivid', 'vouch', 'vouchsafe', 'warm', 'warmer', 'warmhearted', 'warmly', 'warmth', 'wealthy', 'welcome', 'well', 'well-backlit', 'well-balanced', 'well-behaved', 'well-being', 'well-bred', 'well-connected', 'well-educated', 'well-established', 'well-informed', 'well-intentioned', 'well-known', 'well-made', 'well-managed', 'well-mannered', 'well-positioned', 'well-received', 'well-regarded', 'well-rounded', 'well-run', 'well-wishers', 'wellbeing', 'whoa', 'wholeheartedly', 'wholesome', 'whooa', 'whoooa', 'wieldy', 'willing', 'willingly', 'willingness', 'win', 'windfall', 'winnable', 'winner', 'winners', 'winning', 'wins', 'wisdom', 'wise', 'wisely', 'witty', 'won', 'wonder', 'wonderful', 'wonderfully', 'wonderous', 'wonderously', 'wonders', 'wondrous', 'woo', 'work', 'workable', 'worked', 'works', 'world-famous', 'worth', 'worth-while', 'worthiness', 'worthwhile', 'worthy', 'wow', 'wowed', 'wowing', 'wows', 'yay', 'youthful', 'zeal', 'zenith', 'zest', 'zippy', '']

예제3. 아래의 긍정단어가 리스트들 주에 있는지 확인하시오!

```
positive = open("c:₩₩data₩₩positive-words.txt")
pos = positive.read().split('₩n')
if 'wonderful' in pos: #긍정단어가 다 들어있는 pos 리스트에서 wonderful이 존재하면
    print('존재합니다') #존재합니다를 출력합니다.
else:
    print('존재하지 않습니다')
```

결과:
존재합니다

예제4. 스티븐 잡스 연설문(jobs.txt)를 읽어서 한 단어씩 출력하시오!

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split() #어절별로 분리해서 stev2에 저장합니다.
print(stev2) #프린트해보면 stev2가 담겨있습니다
for i in stev2: #stev3리스트에서 하나씩 빼냅니다.
    print(i)
```

■ 87. 문자열에서 대소문자 변환하기(upper, lower)

문자열 객체의 upper() 메소드는 문자열에 있는 모든 알파벳을 대문자로 변환한 결과를 리턴합니다.
문자열 객체의 lower() 메소드는 문자열에 있는 모든 알파벳을 소문자로 변환한 결과를 리턴합니다.

예제:

```
txt = 'A lot of things occur each day'
result1 = txt.upper()
result2 = txt.lower()
print(result1)
print(result2)
```

결과:

```
A LOT OF THINGS OCCUR EACH DAY
a lot of things occur each day
```

■ 88. 문자열에서 좌우 공백 제거하기(lstrip, rstrip, strip)

함수	설명
lstrip	문자열에서 존재하는 왼쪽 공백을 제거
rstrip	문자열에서 존재하는 오른쪽 공백을 제거
strip	문자열에서 존재하는 양쪽 공백을 제거

예제:

```
txt7 = '            양쪽에 공백이 있는 문자열 입니다.            '
print(txt7.lstrip())
print(txt7.rstrip())
print(txt7.strip())
```

결과:

```
양쪽에 공백이 있는 문자열 입니다.
            양쪽에 공백이 있는 문자열 입니다.
양쪽에 공백이 있는 문자열 입니다.
```

■ 89. 문자열에서 문자 개수 구하기 (count)

문자열 객체의 count()메소드는 문자열에서 특정 문자의 개수를 리턴합니다.

```
txt = 'A lot of things occur each day. Today is beautiful day'

print( txt.count('day') ) #txt에서 day단어의 건수를 확인 today포함 3개
```

■ 파이썬에서 막대 그리프 그리는 방법

```
import matplotlib.pyplot as plt
y_value = [21.6, 23.6, 45.8, 77.0]
x_index = [0, 1, 2, 3]

plt.bar(x_index, y_value, color = 'red')  #색깔 바꾸기
plt.show()
```

12/9 (090-106)

2020년 12월 9일 수요일 오전 9:38

■ 파이썬 복습

1. 파이썬 자료형 5가지
2. 파이썬 연산자
3. if문과 loop문
4. 문자열 다루는 함수들
5. 파이썬에서 모듈 사용법
6. 파이썬에서 클래스 사용법
7. 판다스 사용법
8. 예외처리
9. 숫자형 자료를 다루는 함수들
10. 문자열 다루는 함수들2

- 파이썬 자료형 5가지

1. 문자형
2. 숫자형
3. 리스트형
4. 딕셔너리형
5. 튜플형

■ 90. 문자열에서 특정 문자(열) 위치 찾기 (find)

문자열 객체의 find()함수는 문자열에서 특정 문자나 문자열이 위치하는 인덱스를 리턴합니다. find()에서 찾고자 하는 문자를 입력하면 해당 문자나 문자열이 최초로 나타나는 위치에 대한 인덱스를 리턴합니다.

예제:

```
txt = 'A lot of things occur each day'
word_count1 =  txt.find('o')
print(word_count1)  #3이 출력되는데 3은 'o'라는 철자가 위치하는 인덱스 번호를 리턴합니다.

word_count2 = txt.find('day')  #27이 출력됨
print(word_count2)
```

설명: find는 오라클의 instr함수처럼 해당 자리의 인덱스 번호를 출력하는 문자열 함수 입니다.

■ 91. 문자열을 특정 문자(열)로 분리하기(split)

코드를 작성할 때 가장 많이 접하게 되는 경우가 구분자(separator)로 구분되어있는 문자열 파싱(parsing)을 하는 일입니다. 이때 문자열 객체의 split()를 활용하면 구분자를 기준으로 문자열을 쉽게 분리하여 파싱할 수 있습니다.

```
a = 1,"이준혁",29,"남","경영학과"
print(a)   #(1, '이준혁', 29, '남', '경영학과')
------------------
a = '''1,"이준혁",29,"남","경영학과" '''
print(a)   # <class 'str'>
```

result = a.split(',') #콤마(,)로 구분해서 리스트에 담는다.
print(result) # ['1', '"이준혁"', '29', '"남"', '"경영학과" ']

설명: 1,"이준혁",29,"남","경영학과" <--- 이 데이터를 문자열 변수에 할당하고 싶은데 문자열들 중에 더블쿼테이션 마크가 있다면 양쪽에 싱글쿼테이션 마크를 3개를 둘러줘야 합니다.
안그래면 a가 튜플로 만들어짐

- 정리:

1. 문자열.split(',') : 문자열을 콤마(,)로 구분해서 구분하는 문자들을 요소로 해서 a 리스트에 담는다.
2. a = 문자열.split(' ') : 문자열을 공백으로 구분해서 구분하는 문자들을 요소로 해서 a리스트에 담는다.
3. a = 문자열.split() : 문자열을 어절별로 구분해서 구분하는 문자들을 요소로 해서 a리스트에 담는다.

- 웹로그 데이터를 분석하는 선배기수가 모사의 웹사이트에 접속하는 고객들이 해당사이트에서 어느 상품을 많이 클릭을 하는지등을 조회하거나 아니면 웹사이트에서 어느 페이지를 많이 보는지등을 분석하는 업무를한다. (문제291참조)

■ 92. 문자열을 특정 문자(열)로 결합하기(join)

문자열 객채의 'join()메소드는 split()와는 반대로 문자열이 요소인 리스트를 인자로 받아 리스트 모든 요소를 특정 문자열로 연결하여 새로운 문자열로 만들어 리턴합니다.

예제:
a = ['http:', '', 'www.naver.com', 'news', 'today=20191204']
bond = '/'
url = bond.join(a)
print(url)
결과: http://www.naver.com/news/today=20191204

설명: 문자열 함수인 join을 이용해서 리스트의 요소를 불러와서 다시 문자열로 만들 수 있습니다.

■ 93. 문자열에서 특정 문자(열)를 다른 문자(열)로 바꾸기(replace)

문자열 객체의 replace()메소드는 문자열에서 특정 문자나 문자열을 다른 문자나 문자열로 변경합니다.

예제:
txt = 'My password is 1234'
result = txt.replace('1', '0') #txt의 문자열의 숫자1을 숫자0으로 변경해라
print(result)

■ 94. 문자열을 바이트 객체로 바꾸기(encode)

file = open("c:₩₩data₩₩emp1222.csv", encoding="UTF8")

인코딩이란?
인코딩이란 사람이 인지할 수 있는 형태의 데이터를 약속된 규칙에 의해 컴퓨터가 사용하는 0과 1로 변환하는 과정을 말합니다.
컴퓨터 발명, 발전이 미국의 학계와 기업을 중심으로 이루어지다 보니 문자 집합(컴퓨터에서 문자를 담는 방)도 미국을 기준으로 제정되었습니다.
미국에서 제정된 ASCII(미국 정보 교환 표준부호)는 1960년대에 제정된 문자집합으로 이후 개발된 문자 집합의 토대를 이루고 있습니다.
ASCII는 7비트만을 이용해서 음이 아닌수(0~127)의 문자집합내의 문자 할당 약속입니다.

예제: A는 숫자 65입니다. a는 숫자 97입니다.

구글에서 아스키 코드표를 검색해보세요~~

예제: chr함수를 이용해서 숫자 65부터 127까지가 어떤 문자인지 출력하시오!

```
for i in range(65, 128):
        print(i, '----> ', chr(i) )
```

결과:

```
65 ----> A
66 ----> B
67 ----> C
68 ----> D
69 ----> E
70 ----> F
71 ----> G
72 ----> H
73 ----> I
74 ----> J
75 ----> K
76 ----> L
77 ----> M
78 ----> N
79 ----> O
80 ----> P
81 ----> Q
82 ----> R
83 ----> S
84 ----> T
85 ----> U
86 ----> V
87 ----> W
88 ----> X
89 ----> Y
90 ----> Z
91 ----> [
92 ----> ₩
93 ----> ]
94 ----> ^
```

95 ----> _
96 ----> `
97 ----> a
98 ----> b
99 ----> c
100 ----> d
101 ----> e
102 ----> f
103 ----> g
104 ----> h
105 ----> i
106 ----> j
107 ----> k
108 ----> l
109 ----> m
110 ----> n
111 ----> o
112 ----> p
113 ----> q
114 ----> r
115 ----> s
116 ----> t
117 ----> u
118 ----> v
119 ----> w
120 ----> x
121 ----> y
122 ----> z
123 ----> {
124 ----> |
125 ----> }
126 ----> ~
127 ----> □

이 아스키 코드표로 미국 엔지니어들은 행복하게 살아가고 있었지만 다른나라 사람들은 그러지 못했습니다.

1. 서유럽 국가
   독일어의 움라우트, 스페인어의 물결표, 터키어의 시딜러, 프랑스어의 악센트는 7비트로는 표현 못합니다.

2. 아시아
   아시아에서는 한자, 일본어, 한글이 7비트에 들어맞지가 않았습니다. 더 큰 문자셋이 필요했는데 16비트로 된 문자집합을 제공해서 해결했습니다.
   그 문자 집합셋이 바로 CP949, EUC-KR 입니다.
   file = open('c:₩₩data₩₩emp1222.csv', encoding="CP949")

   포스코 더 샆 아파트에서 사는 사람들의 주소를 컴퓨터에 입력할 때 어떻게 보이냐면 '포스코 더 ? 아

파트'로 보였다.
그래서 해결한게 바로 유니코드 문자 집합셋입니다.
바로 그게 UTF8, UTF16입니다.
file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")

파이썬에서 한글이 ??로 출력이 되면 UTF8로 인코딩하면 됩니다.

■ 95. 바이트 객체를 문자열로 바꾸기(decode)

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")

인코딩(encoding) ?
사람이 알아볼 수 있는 언어 ---------->컴퓨터가 알아볼 수 있는 언어(숫자)

디코딩(decoding) ?
컴퓨터가 알아볼 수 있는 언어 ---------> 사람이 알아볼 수 있는 언어

예제:
```
txt = 'A'
b_txt = txt.encode() #txt 변수에 들어있는 대문자 A를 인코딩한다.
print(b_txt) #b'A' --> binary'A'라는 뜻
                      (이진) --> 0과 1로 표현하는 데이터

c_text = b_txt.decode() #컴퓨터가 알아볼 수 있는 언어 --> 사람이 알아볼 수 있는 언어로 변경
print(c_txt) #A

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
```

설명: emp1222.csv를 파이썬으로 로드하는데 파이썬으로 load하려면 컴퓨터가 알아볼 수 있는 언어로 변경해야 합니다.
이때 문자집합셋을 유니코드 문자집합셋인 UTF8을 이용해서 인코딩 즉 컴퓨터가 알아볼 수 있는 언어로 변경하겠다는 것입니다. (ANSI: ASCII를 담는컵, ANSI보다 큰 컵이 UTF8)

■ 96. 순차적인 정수 리스트 만들기(range)

[0, 1, 2, 3]이나 [100, 101, 102, 103]과 같이 순차적인 정수 리스트를 만드는 가장 간단한 방법은 파이썬 내장함수인 range()를 이용하는 것입니다.

예제:
```
print( range(1,11) ) #range(1,11)
print(list(range(1,11)))  #[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

■ 97. 리스트에서 특정 위치의 요소 얻기

리스트의 특정 위치에 있는 요소값을 구하려면 인덱싱을 이용합니다.
인덱스는 0부터 시작합니다.

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6] ]
print(a)
```

예제1. 위의 리스트 a에서 숫자2를 출력하시오!

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6] ]
print(a[1])
```

예제2. 위의 리스트 a에서 숫자4를 출력하시오!

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6] ]
print(a[5])   #[4, 5, 6]
print(a[5][0]) #4 ---> 리스트 안에 있는 리스트의 특정 요소를 뽑을 땐 이렇게 한다.
```

■ 98. 리스트에서 특정 요소의 위치 구하기(index)

리스트 객체의 index()메소드는 리스트에서 요소의 값을 알고 있을 때 그 요소가 최초로 나타나는 위치의 인덱스를 리턴합니다.

예제:

```
a = [1, 2,'a', 'b', 'c', [1, 2, 3]]
print( a.index(2) ) #1
print( a.index('a') ) #2
```

■ 99. 리스트에서 특정 위치의 요소를 변경하기

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print( list_a[0] ) #a
list_a[0] = 'z'
print(list_a) #['z', 'b', 'c', 'd', 'e', 'f', 'g']
```

■ 100. 리스트에서 특정 구간에 있는 요소 추출하기

리스트에서 특정 구간에 있는 요소를 추출하여 새로운 리스트를 만들려면 슬라이싱을 이용하면 됩니다.

예제:

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(list_a[0:4]) #['a', 'b', 'c', 'd']
print(list_a[2:]) #['c', 'd', 'e', 'f', 'g'] 리스트의 인덱스 2번째 요소부터 읽어서 끝까지 읽어라
print(list_a[:3]) #['a', 'b', 'c'] 리스트의 처음부터 읽어서 인덱스 번호부터 3미만까지 읽어라
```

■ 101. 리스트에서 짝수 번째 요소만 추출하기

리스트에서 짝수번째 요소를 추출하는 방법은 두번째 요소부터 끝까지 스텝2로 슬라이싱 하면 됩니다.

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
print(list_a[1 : : 2]) #인덱스 번호 1번째 요소부터 끝까지 2스텝으로 읽어라 (두칸씩 건너뛰면서 읽어라)
```

결과:
['b', 'd', 'f', 'h']

■ 102. 리스트 요소 순서를 역순으로 만들기 ① (reverse)

리스트 객체의 reverse() 메소드는 리스트의 모든 요소 순서를 거꾸로 만듭니다.

예제:

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
list_a.reverse()
print(list_a) #['g', 'f', 'e', 'd', 'c', 'b', 'a']
```

■103. 리스트 요소 순서를 역순으로 만들기 ② (reversed)

리스트 요소 순서를 역순으로 만드는 또 다른 방법은 파이썬 내장 함수인 reversed() 이용하는 방법입니다. reverse()는 인자로 입력된 시퀀스 자료형의 요소 순서를 역순으로 새로운 시퀀스 자료형을 만들어 리턴합니다.

reverse() 함수 : 원본 데이터를 역순으로 만듭니다.
reversed() 함수 : 역순 데이터를 만들지만 원본 데이터를 역순으로 만들지 않습니다.

예제:

```
list_a = ['a', 'b', 'c', 'd', 'e']
list_a.reverse() #list_a의 요소를 역순으로 변경한다.
print(list_a) #['e', 'd', 'c', 'b', 'a']

list_b = ['a', 'b', 'c', 'd', 'e']
result = reversed(list_b) #list_b의 요소를 역순으로 만들어서 result에 저장
print(list(result)) #['e', 'd', 'c', 'b', 'a']
print(list_b) #['a', 'b', 'c', 'd', 'e']
```

■ 104. 리스트 합치기( )

두개의 리스트를 연결하여 새로운 리스트를 만드는 방법은 + 연산자를 이용해 두개의 리스트를 더하면 됩니다.

예제:

listdata1 ['a', 'b', 'c', 'd']

```
listdata1 = ['a', 'b', 'c', 'd']
listdata2 = ['f', 'g', 'h', 'i']
listdata3 = listdata1 + listdata2
print(listdata3)        #['a', 'b', 'c', 'd', 'f', 'g', 'h', 'i']
```

■ 105. 리스트 반복하기(*)

list*n은 리스트를 n번 반복해서 새로운 리스트를 만듭니다.
예를들어, list*2 는 list + list와 동일한 결과가 출력됩니다.

예제:

```
listdata = list( range(3) )
result = listdata * 3
print(result) #[0, 1, 2, 0, 1, 2, 0, 1, 2]
```

■ 106. 리스트에 요소 추가하기(append)

리스트 객체의 append() 메소드는 인자로 입력된 값을 리스트의 맨 마지막 요소로 추가합니다.

예제:

```
listdata = []
listdata.append('a')
listdata.append('b')
print(listdata) #['a', 'b']
```

■ 파이썬으로 히스토그램 그래프 그리기

계급을 가로축에 도수를 세로축에 나타낸 뒤 각 계급의 크기를 가로의 길이로 하고 도수를 세로의 길이로 하는 직사각형을 차례대로 그려서 나타낸 그래프

예제:

1. 평균이 150이고 표준편차가 5인 초등학생 10만명의 키를 생성한다.

   ```
   import numpy as np
   height = np.random.randn(100000) * 5 + 150
   ```

2. 계급의 크기를 나타내는 가로의 길이를 설정

   ```
   bins = [142, 144, 146, 148, 150, 152, 154, 156, 158, 160 ]
   ```

3. 히스토그램 그래프의 세로축에 해당하는 도수(건수)를 구하는 코드

   ```
   hist, bins = np.histogram( height, bins)
   print(hist, bins)
   ```

   결과:

   ```
   [ 6012  9459 13292 15440 15705 13264  9774  6002  3203]
   ```

[142 144 146 148 150 152 154 156 158 160]

142~144미만 : 6012명
144~146미만 : 9459명
: :
159~160미만 : 3203명

4. 위의 데이터를 가지고 히스토그램 그래프를 그린다.

```
import matplotlib.pyplot as plt

plt.hist(hist, bins)
```

```
>>>
import numpy as np
import matplotlib.pyplot as plt

height = np.random.randn(100000) * 5 + 150
bins = [142, 144, 146, 148, 150, 152, 154, 156, 158, 160 ] #계급

hist, bins = np.histogram( height, bins) #도수 확인하려고 한번 본 데이터
plt.grid() #격자모양이 생김
plt.hist(height, bins, rwidth = 0.9, alpha = 0.7, color = 'red')
```

↑
초등학생 10만명의 키 데이터

rwidth --> 히스토그램 그래프의 넓이
alpha --> 색깔투명도

(눈금조절 방법)

```
plt.xticks(x_val)          # x축 눈금 조절. 검색해서 찾았습니다.
plt.yticks(range(0,12))    # y축 눈금 조절.
```

12/10 (107-117)

2020년 12월 10일 목요일 오전 10:01

■ 107. 리스트의 특정 위치에 요소 삽입하기(insert)

리스트 객체의 insert() 메소드는 리스트 특정 위치의 새로운 요소를 입력합니다.

예제:

```
list_a = ['a', 'b', 'c', 'd', 'e']
list_a.append('f') #append 메소드는 무조건 맨 뒤쪽에 요소를 추가한다.
print(list_a) #['a', 'b', 'c', 'd', 'e', 'f']

list_a.insert(3, 'k')
print(list_a) #['a', 'b', 'c', 'k', 'd', 'e', 'f']
```

설명: list_a.insert(3, 'k')는 list_a 리스트의 인덱스번호 3번으로 k를 구성해라 라는 뜻

■ 108. 리스트의 특정 위치의 요소 제거하기(del)

- 파이썬의 자료형(자료구조) 5가지의 해당 메소드와 if문과 loop문

1. 문자형 count, index,....
2. 숫자형 int(), round() + if문과 loop문
3. 리스트형 count, append, insert, del, ... + 함수생성, 클래스, 예외처리
4. 튜플형
5. 사전형

※ del 메소드와 remove 메소드의 차이?

둘다 리스트안의 요소를 삭제하는 것인데 remove는 요소명으로 삭제하는 것이고 del은 리스트의 인덱스 번호로 삭제하는 것입니다.

```
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']

del(a[4]) # a 리스트에 인덱스 번호 4번의 해당하는 요소를 삭제한다.
print(a)
```

■ 109. 리스트에서 특정 요소 제거하기(remove)

리스트에서 특정 요소의 값을 알고 있을 때 그 요소를 제거하는 방법은 리스트 객체의 remove() 메소드를 이용하면 됩니다.

예제:
```
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']
a.remove('목성')
print(a)
```

■ 110. 리스트에서 특정 구간에 있는 모든 요소 제거하기

리스트 객체에서 특정 구간에 있는 요소들을 제거하려면 del과 술라이싱을 이용하면 됩니다.

예제:
```
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']
del(a[2:6]) #인덱스번호 2번부터 6번 미만까지 요소를 지움
print(a)
```

■ 111. 리스트에 있는 요소 개수 구하기(len)

파이썬 내장함수인 len()은 시퀀스 자료형의 크기를 구하는 함수입니다.
len()을 리스트에 적용하면 리스트의 모든 요소의 개수를 리턴합니다.

예:
```
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']
print(len(a)) #9
```

■ 112. 리스트에서 특정 요소 개수 구하기(count)

리스트.count('요소')는 객체 리스트에서 '요소'의 개수를 구하는 메소드입니다.

```
box = ['yellow', 'yellow', 'red', 'red', 'red']
print(box.count('red')) #3
```

■ 113. 리스트 제거하기(del)

리스트 자체를 메모리에서 제거하려면 다음과 같이 del명령어를 수행합니다.

예제:
```
a = [1,2,3,3,3,4]
del a
print(a)
```

결과: NameError: name 'a' is not defined (에러남) <--- 메모리에서 완전히 지우는 명령어

설명 : 스파이더 오른쪽 아래에 있는 지우개 기능은 현재까지 스파이더를 사용하면서 정의된 모든 변수를 다 지우는 기능입니다.

■ 114. 리스트 요소 정렬하기 ① (sort)

list.sort()는 리스트 객체 list의 모든 요소를 정렬합니다.

얘 :
```
a = [2, 4, 1, 3, 5]
a.sort()
print(a) #[1, 2, 3, 4, 5]
```

만약에 역순으로 정렬하고 싶다면
```
a = [2, 4, 1, 3, 5]
a.sort(reverse = True)
print(a)  #[5, 4, 3, 2, 1]
```

■ 115. 리스트 요소 정렬하기 ② (sorted)

파이썬 내장함수 sorted()를 이용하여 리스트를 정렬할 수도 있습니다. 리스트 객체의 sort() 메소드와 다른점은 리스트 객체의 sort()는 원본 리스트를 정렬한 형태로 변경하지만 sorted()는 원본 리스트는 그대로 두고 정렬한 결과 리스트만 리턴합니다.

예제:
```
a = [2,1,3,5,4]
a.sort()
print(a)
----------
a = [2,1,3,5,4]
b = sorted(a)

print(a) #[2, 1, 3, 5, 4] 원본은 그대로 둡니다.
print(b) #[1, 2, 3, 4, 5]
```

■ 116. 리스트 요소 무작위로 섞기(shuffle)

파이썬 기본 모듈인 random은 난수를 발생하는 모듈입니다.
이 모듈이 제공하는 shuffle은 리스트의 요소를 무작위로 섞는데 활용됩니다.

예제:
```
from random import shuffle

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
shuffle(a) #[7, 4, 3, 5, 1, 6, 2, 9, 10, 8] / 수행할 때마다 결과 달라짐
```

print(a)

■ 117 리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate)

파이썬 내장함수 enumerate()는 시퀀스 자료형을 인자로 받아 각 요소를 인덱스와 함께 쌍으로 추출할 수 있는 반복가능한 자료인 enumerate 객체를 리턴합니다. enumerate 객체는 주로 for문과 함께 자주 사용되고 list()를 이용해서 리스트 객체를 변환할 수 있습니다.

예제:

```
emp12 = ['김주원', '김미승', '김홍비', '권세원']
result = list( enumerate(emp12) )
print(result)
```

```
[(0, '김주원'), (1, '김미승'), (2, '김홍비'), (3, '권세원')]
```

예제2:

```
emp12 = ['김주원', '김미승', '김홍비', '권세원']
for i, k in enumerate(emp12):
    print(i,k)
```

```
0 김주원
1 김미승
2 김홍비
3 권세원
```

12/14 (118-131)

2020년 12월 14일 월요일 오전 9:56

■ 118. 리스트의 모든 요소의 합 구하기(sum)

파이썬 내장 함수 sumn()을 이용해 list의 모든 요소의 합을 출력합니다.

예제:

```
listdata = [ 2, 2, 1, 3, 8, 4, 3, 9, 2, 20]
result = sum(listdata)
print(result)   #54
```

■ 119. 리스트 요소가 모두 참인지 확인하기(all, any)

리스트의 모든 요소가 참인지 또는 모든 요소가 거짓인지 판단해야 하는 경우 파이썬 내장 함수 all()이나 any()를 사용하면 됩니다.
all()은 인자로 입력되는 리스트의 모든 요소가 참인 경우에만 True를 리턴하고 거짓이 하나라도 포함되어져 있으면 False를 리턴합니다.
이와는 달리 any()는 인자로 입력되는 리스트의 모든 요소가 거짓인 경우에만 False를 리턴하고 참이 하나라도 존재하면 True를 리턴합니다.

예:

```
listdata1 = [True, True, True]
listdata2 = [True, False, True]
print( all(listdata1) ) #True
print( all(listdata2) ) #False
print( any(listdata1) ) #True
print( any(listdata2) ) #True

--------------------------
listdata1 = [1, 1, 1]
listdata2 = [1, 0, 1]
print( all(listdata1) ) #True
print( all(listdata2) ) #False
print( any(listdata1) ) #True
print( any(listdata2) ) #True
```

설명 : True는 숫자로는 1이고 False는 숫자로는 0입니다.

■ 120. 사전에 요소 추가하기

파이썬 책 큰 목차:

1. 문자형과 문자열 함수들
2. 숫자형과 숫자형 함수들 + if과 loop문
3. 리스트형과 리스트형 함수들 + 모듈과 클래스
4. 사전형과 사전형 함수들 + 예외처리

+

판다스 사용법, 웹스크롤링 <----- 데이터 분석가 필수 기술

사전형은 키:값으로 되어진 요소로 구성되어 있습니다.
사전형은 리스트형처럼 인덱스번호로 요소에 접근하는게 아니라 키값으로 요소의 값에 접근합니다.

예제:

```
sol = { } #중괄호를 써서 비어있는 딕셔너리를 생성합니다.
sol['태양'] = 'sun'
print(sol)   #{'태양': 'sun'}
sol['수성'] = 'mercury'
sol['금성'] = 'venus'
print(sol) #{'태양': 'sun', '수성': 'mercury', '금성': 'venus'}
```

■ 121. 사전의 특정 요소값 변경하기

사전 자료형의 특정 요소값을 변경하는 방법은 다음과 같습니다.

예제: 사전형이름['키값'] = '값'

```
sol['태양']  'aaaaaa'
print(sol)
```

■ 122. 사전의 특정 요소 제거하기(del)

사전 dict에서 특정요소 key:value를 제거하는 방법은 다음과 같습니다.

예제:

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', 'mercury': '수성', 'venus': '금성', 'earth': '지구', 'mars': '화성'}
del sol['태양']
print(sol) #{'수성': 'mercury', '금성': 'venus', 'mercury': '수성', 'venus': '금성', 'earth': '지구', 'mars': '화성'}
```

■ 123. 사전의 모든 요소 제거하기(clear)

사전의 모든 요소를 제거하여 빈 사전으로 만드는 방법은 사전 객체의 clear() 메소드를 이용하면 됩니다.

예제:
```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
dict.clear()
print(dict)    #{}
```

■ 124. 사전에서 키만 추출하기(keys)

아래의 사전에서 key만 추출하고 싶으면 사전이름.keys()라고 하면 됩니다.

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
print(sol.keys())    #dict_keys(['태양', '수성', '금성', '지구', '화성'])
```

■ 125. 사전에서 값만 추출하기(values)

아래의 사전형에서 값에 해당하는 부분만 추출하려면 사전형이름.values()

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
print( sol.values() ) #dict_values(['sun', 'mercury', 'venus', 'earth', 'mars'])
```

■ 126. 사전 요소를 모두 추출하기(items)20130 사전에 특정 키가 존재하는지 확인하기(in)

딕셔너리 자료형에서 데이터를 추출하는 방법은 다음과 같습니다.

1. key만 추출: dict.keys()
2. 값만 추출: dict.values()
3. 둘다 추출: dict.items()

예제:
```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

print(gini.keys())
print(gini.values())
print(gini.items())
#dict_items([('비틀즈', ['yesterday', 'imagine']), ('아이유', ['너랑나', '마슈멜로우']), ('마이클 잭슨', ['beat it', 'smooth criminal'])])
```

■ 127. 사전 정렬하기(sorted)

파이썬 내장함수 sorted()는 사전 자료를 인자로 입력받아 정렬할 수 있습니다.
사전의 각 요소는 키:값 으로 되어져 있습니다.
sorted()함수에 사전을 인자로 입력하면 기본적으로 사전의 키를 오름차순으로 정렬한 결과를 리스트로 반환합니다.

예제:
dict2 = ['소녀시대' : '소원을 말해봐', '방탄 소년단':'DNA', '오마이걸' : '살짝 설렜어']

# 1. 위의 dict2에서 키만 추출
dict2 = {'소녀시대':'소원을 말해봐', '방탄 소년단':'DNA', '오마이걸':'살짝 설렜어'}
print(dict2.keys())

결과 : dict_keys(['소녀시대', '방탄 소년단', '오마이걸'])

# 2. 위의 dict2의 키값을 아래와 같이 정렬해서 출력하시오!
dict2 = {'소녀시대':'소원을 말해봐', '방탄 소년단':'DNA', '오마이걸':'살짝 설렜어'}
result = dict2.keys()
print(sorted(result))

결과 : ['방탄 소년단', '소녀시대', '오마이걸']

* defaultdictionary 를 이용해서 딕셔너리의 값을 리스트로 구성하는 코드

```
from collections import defaultdict
gini = defaultdict(list)
gini['kt'].append('허혁')
gini['kt'].append('정다희')
gini['sk'].append('박혜진')
gini['sk'].append('김승순')
print( gini )
```

결과:
defaultdict(<class 'list'>, {'kt': ['허혁', '정다희'], 'sk': ['박혜진', '김승순']})

■ 128. 문자 코드값 구하기(ord)

파이썬 내장함수 ord()는 문자를 컴퓨터가 인식하는 코드값으로 변환합니다.

예제 : A ----------------> 65
인코딩
↓
사람이 알아볼 수 있는 언어를 컴퓨터가 알아볼 수 있는 숫자로 변환

```
print( ord('A') ) #65
```

■ 129. 코드값에 대응하는 문자 얻기(chr)

파이썬 내장함수 chr()은 ord() 함수의 반대기능을 수행합니다.

chr()의 인자로 정수값을 입력하면 이 정수값이 해당하는 문자를 리턴합니다.

예제:
print(chr(65)) #A

■ 130. 문자열로 된 식을 실행하기(eval)

코드를 작성하다 보면 파일에서 읽은 수식이나 문자열을 그대로 실행해야하는 경우가 있습니다. 예를 들어 '2+3'과 같이 두개의 숫자를 더하는 문자열을 텍스트 파일에서 읽어 이를 실행하여 이 수식의 결과인 5를 구하는 경우입니다.

예제:
```
a = '2+3'
print(a) #2+3
print(eval(a)) #5
```

■ 131. 이름없는 한줄짜리 함수 만들기(lambda)

파이썬에서 함수를 정의하는 방법은 일반적으로 다음과 같습니다.

```
def 함수이름(입력매개변수):
        실행코드
```

예제 :
```
def add_func(a,b):
    return a + b

print(add_func(2,3)) #5
```

설명 : 함수를 만들 때 위와같이 함수 이름을 주고 생성했습니다.
함수를 생성하는 이유는 함수를 한번 만들면 다음부터는 함수이름을 가지고 함수만 호출하면 되기 때문에 함수를 생성해서 코드를 작성했습니다.

지금부터 만들 함수는 함수 이름 없이 함수를 생성하는 코드입니다.
지금 이 코드에서 한번만 실행하고 앞으로 쓸 일이 없는 경우 함수 이름없이 함수를 생성합니다.

아래와 같이 한 줄짜리 함수를 생성합니다.

문법: 변수이름 = lambda 입력매개변수 : 실행문

```
k = lambda a, b : a+b #lamda
print( k(2,3) ) #5
```

<신뢰구간>

신뢰구간 68% -------> 평균 ± 1 + 표준편차

신뢰구간 95% -------> 평균 ± 1.96 + 표준편차

신뢰구간 99% -------> 평균 ± 2.58 + 표준편차

12/15 (132-141)

2020년 12월 15일 화요일 오전 9:47

■ 132 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map)

파이썬 내장함수 map은 리스트 A와 함수 f가 주어지면 리스트 A의 요소를 함수 f에 입력해서 출력하는 결과를 간단하게 출력해줍니다.

문법: map(함수명, 함수에 제공할 매개변수 값들)

매개변수의 값들을 바꿔가면서 함수를 반복실행할 때 유용하게 사용할 수 있습니다.

예제:

```
def gob(x):
        return x*x

a = [1,2,3,4,5]
result = map(gob, a) #map(함수명, 리스트명)
print( list(result) )  #[1, 4, 9, 16, 25]
```

설명:

- map함수가 리턴해주는 결과를 보려면 반드시 list함수로 리턴된 값을 변환해서 리스트 형태로 봐야합니다. 안 그러면 결과가 <map object at 0x000002191CA5EB80> 이렇게 나옴
- 리스트의 요소 하나하나를 함수의 입력값으로 입력한다(mapping)

■ 133 텍스트 파일을 읽고 출력하기(read)

텍스트 파일을 읽고 그 내용을 화면에 출력하고자 하면 제일 먼저 텍스트 읽기 모드로 파일을 엽니다.
텍스트 파일을 오픈하면 텍스트 파일을 읽어 내용을 화면에 출력하면 됩니다.

예:

```
f = open("c:₩₩data₩₩jobs.txt")
data = f.read() #파일을 한번에 전부 읽어오는 함수 입니다.
print(data)
f.close()
```

설명: 스티븐 잡스 연설문은 길지 않으니까 한번에 읽어와도 관계 없는데 텍스트 파일의 용량이 매우 클 경우 read()로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기시킬 수 있습니다.

웹에서 스크롤링한 데이터를 분석하고자 할 때 위와 같이 open 함수를 이용해서 파이썬으로 데이터 불러옵니다.

■ 134. 텍스트 파일을 한줄씩 읽고 출력하기 ① (readline)

텍스트 파일의 용량이 매우 클 경우 read()로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기할 수 있습니다.
이 경우 텍스트 파일 내용을 한 줄 단위로 읽고 작업을 수행하면 됩니다.
readline()은 텍스트 파일에서 한 줄을 읽습니다.
한 줄을 읽고 나면 파일을 읽기 시작하는 위치는 그 다음줄의 맨 처음이 됩니다.

예제:

```
f = open("c:₩₩data₩₩jobs.txt")
data = f.readline()
print(data)
f.close()
```

설명: read() 함수는 텍스트 파일 전체를 읽어오는 반면 readline()은 텍스트 파일에서 한줄만 읽어옵니다.

■ 135. 화면에서 사용자 입력을 받고 파일로 쓰기(write)

사용자로부터 입력받은 텍스트를 파일로 저장하려면 파일을 텍스트 쓰기 모드로 오픈하고 파일 객체의 write()를 이용해 데이터를 파일에 기록합니다.

예제:

```
text = input('파일에 저장할 내용을 입력하세요~')
f = open('c:₩₩data₩₩mydata.txt', 'w')
f.write( text )
f.close()
```

설명: 위에 w는 write의 약자입니다. c드라이브 밑에 data밑에 mydata.txt파일로 쓰겠다는 뜻입니다.

파일에 저장할 내용을 입력하세요~ 나는 데이터 분석가로 일하게 됩니다.
---> mydata파일로 저장됨

설명 : 위의 코드는 웹에서 스크롤링한 데이터를 파일로 저장할 때 응용되게 됩니다.

■ 136. 텍스트 파일에 한줄씩 쓰기(writelines)

파일 객체의 writelines()는 텍스트 문자열이나 텍스트 문자열이 요소로 되어있는 리스트를 인자로 받아 파일에 한 줄씩 기록합니다. 리스트가 인자인 경우 writelines()는 리스트 요소를 하나의 문자열로 결합하여 파일을 한꺼번에 기록합니다.

※ writelines는 리스트 자료형도 파일에 저장할 수 있습니다.

예제:

```
listdata = [2, 2, 1, 3, 8, 5, 7]
result = sorted( listdata )
```

```
print(result) #[1, 2, 2, 3, 5, 7, 8]
f = open("c:₩₩data₩₩mydata2.txt", "w") #mydata2.txt를 생성하겠다.
f.write( str(result) ) #result에 있는 내용을 mydata2.txt로 생성한다.
f.close() #result에 있는 내용을 문자로 변환해야 한다.

---------------------------------
listdata = [2, 2, 1, 3, 8, 5, 7]
result = sorted( listdata )
print(result) #[1, 2, 2, 3, 5, 7, 8]
f = open("c:₩₩data₩₩mydata22.txt", "w") #mydata2.txt를 생성하겠다.
f.write( result ) #result에 있는 내용을 mydata2.txt로 생성한다.
f.close()
```

그냥 이렇게 쓰면 TypeError: write() argument must be str, not list 라는 오류가 난다.

writelines 예제:

```
data = [ ] #data라는 비어있는 리스트 생성
while True: #무한 루프를 수행합니다.
        text = input( '저장할 내용을 입력하세요 ')
        if text == '':   #text에 아무것도 입력하지 않으면
                break    #break문을 실행해서 무한루프를 종료합니다.
        data.append( text + '₩n' ) #text가 엔터와 함께 data리스트에 append됩니다.

f = open('c:₩₩data₩₩mydata99.txt', 'w') #mydata99.txt를 생성하겠다.
f.writelines(data)   #data리스트의 내용을 mydata99.txt에 저장하겠다.
f.close( )

-------------------------------------------------------
data = [ ]
while True:
        text = input( '저장할 내용을 입력하세요 ')
        if text == '':
                break
        data.append( text + '₩n' )

f = open('c:₩₩data₩₩mydata99.txt', 'a')
f.writelines(data)
f.close( )
```

설명: open의 옵션중 w는 그냥 쓰기이고 a는 추가입니다.

■ 137. 텍스트 파일 복사하기(read, write)

텍스트 파일을 복사하는 방법을 파이썬으로 구현하려면 read와 write를 이용하면 됩니다. read()로 텍스트 파일

을 읽고 write로 읽을 내용을 다른 파일이름으로 저장하면 됩니다.

예제:

```
f = open("c:₩₩data₩₩mydata.txt", "r") #r은 read입니다
h = open("c:₩₩data₩₩mydata_copy.txt", "w") #w는 write입니다.
data = f.read()
h.write( data )
f.close()
h.close()
```

■ 138. 바이너리 파일 복사하기(read, write)

이미지나 동영상도 파이썬으로 복사 붙여넣기를 할 수 있습니다.
이미지나 동영상을 복사 붙여넣기 할 때는 파일의 크기가 크므로 한번에 읽어들일 수는 없고 일부분만 일정한 크기로 읽으면서 복사 붙여넣기를 합니다.

예제:

```
bufsize = 1024
f = open("c:₩₩data₩₩lena.png", "rb") #rb는 read binary의 약자입니다.
h = open("c:₩₩data₩₩lena_copy.png", "wb") #wb는 write binary의 약자
data = f.read(bufsize) # 이미지를 1kb을 읽어서 data에 저장합니다.
while data: #data에 data가 발견되는 동안에 루프문을 수행합니다.
    h.write(data) #lena_copy.png에 1kb의 데이터씩 write합니다.
    data = f.read(bufsize) #lena.png에서 1kb의 데이터를 read합니다.
f.close()
h.close()
```

설명: lena.png의 총 크기가 334kb이므로 1kb씩 읽어들여서 쓰는 작업을 총 334번 수행합니다.

■ 139. 파일을 열고 자동으로 닫기 (with ~ as)

with ~ as절을 사용하게 되면 f.close()를 명시하지 않아도 되므로 프로그래머가 실수로 f.close()를 작성하지 않아서 발생하는 메모리 부족 오류를 예방할 수 있습니다.

1. with ~ as절을 사용 안했을 때

```
f = open("c:₩₩data₩₩mydata.txt", "r")
data = f.readlines()
print(data)
f.close() #[' 나는 데이터 분석가로 일하게 됩니다.']
```

2. with ~ as절을 사용했을 때

```
with open("c:₩₩data₩₩mydata.txt", "r") as f:
        data = f.readlines()
        print( data ) #[' 나는 데이터 분석가로 일하게 됩니다.']
```

설명: with ~ as절을 사용하면 자동으로 파일 close를 해줍니다.

■ 140. HTML 기본문법

⊙ 데이터 분석 순서

데이터 수집 --> 데이터 유형 및 속성 파악 ---> 데이터 변환 --> 데이터 저장 -->데이터 정제 --> 데이터 분석

⊙ 데이터 수집 기술
- 웹스크롤링 기술

⊙ html이란? Hyper Text Markup Language의 약자이고 여러개의 태그(tag)를 연결해서 모아놓은 문서

예제1. 아주 간단한 html문서를 만들어 봅니다.

메모장을 열고 아래와 같이 코딩을 하세요~

```
<html><head><title> 양건준님의 오늘 일정 </title></head> 본문의 타이틀이 된 부분
<body>
<p class="title">양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </p> 바디의 타이틀이 된 부분
</body>
</html>
```

메모장의 파일이름을 a.html로 바탕화면에 저장하세요
저장할 때 파일형식을 모든파일로 합니다.

탭 부분에 제목이 들어가고 밑 내용이 들어간 것을 알 수 있다.
시작부분에<head> 해줬으면 끝부분에는 </head> 이런식으로 슬래쉬를 넣어줘야하는 것을 알 수 있다.

※ 웹 페이지 오픈된 상태에서 오른쪽 마우스 클릭 후 페이지 소스보기를 클릭하면 페이지의 소스를 볼 수 있습니다

예제2. 예제1의 글씨를 진하게 하시오!

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"> <b> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </b> </p>
</body>
</html>
```

설명: 글씨를 진하게 하려면 b테그를 사용합니다.

예제3. 예제2의 글씨에 밑줄을 그어 보시오~

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"><b><u> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </u></b></p>
</body>
</html>
```

설명: 밑줄을 사용하려면 테그 u를 사용합니다.

예제4. 예제3의 글씨를 이텔릭체로 변경하시오!

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u></b></p>
</body>
</html>
```

설명: 글씨를 이텔릭체로 변경하려면 테그 i 를 이용합니다.

예제5. p테그를 추가해서 제목과 내용을 나누시오!

<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<u> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </u>
양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육탕으 먹었습니다.
코로나에대한 걱정도 있었지만 용기내서 먹었습니다
</body>
</html>

설명: 제목은 class title에 적고 내용은 class content에 적었습니다.

예제6. 예제5에서 만든 html 문서 본문에 링크를 거시오!
(링크는 우리반 카페주소로 하세요~)

<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<u> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </u>
양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육탕으 먹었습니다.
코로나에대한 걱정도 있었지만 용기내서 먹었습니다

 다음카페

</body>
</html>

■ 141. beautiful soup 모듈 배우기

⊙ beautiful soup 모듈이란? 파이썬 코드를 복잡하게 작성하지 않아도 편하게 웹스크롤링을 할 수 있도록 여러 함수들을 제공하는 웹스크롤링 전문 모듈입니다.

⊙ 카페에서 ecologicalpyramid.html 문서를 내려 받아 c 드라이브 밑에 data폴더 밑에 저장하세요~

예제1. ecologicalpyramid.html 코드를 beautiful soup모듈에서 사용할 수 있도록 변환하고 파싱을 하고 파싱된 내용을 출력하시오!

```
from bs4 import BeautifulSoup

f = open("c:₩₩data₩₩ecologicalpyramid.html")
soup = BeautifulSoup( f, "html.parser")
print(soup)
```

결과:

```
<html>
<body>
<div class="ecopyramid">
<ul id="producers">
<li class="producerlist">
<div class="name">plants</div>
<div class="number">100000</div>
</li>
<li class="producerlist">
<div class="name">algae</div>
<div class="number">100000</div>
</li>
</ul>
<ul id="primaryconsumers">
<li class="primaryconsumerlist">
<div class="name">deer</div>
<div class="number">1000</div>
</li>
<li class="primaryconsumerlist">
<div class="name">rabbit</div>
<div class="number">2000</div>
</li>
</ul>
<ul id="secondaryconsumers">
<li class="secondaryconsumerlist">
<div class="name">fox</div>
<div class="number">100</div>
</li>
<li class="secondaryconsumerlist">
<div class="name">bear</div>
```

```
<div class="number">100</div>
</li>
</ul>
<ul id="tertiaryconsumers">
<li class="tertiaryconsumerlist">
<div class="name">lion</div>
<div class="number">80</div>
</li>
<li class="tertiaryconsumerlist">
<div class="name">tiger</div>
<div class="number">50</div>
</li>
</ul>
</div></body>
</html>
```

예제2. ecologicalpyramid.html 코드에서 class 이름 name에 접근해서 데이터를 긁어오시오!

```
f = open("c:₩₩data₩₩ecologicalpyramid.html")
soup = BeautifulSoup( f, "html.parser")
result = soup.find_all(class_ ="name")
print(result)
```

결과:

```
[<div class="name">plants</div>,
<div class="name">algae</div>,
<div class="name">deer</div>,
<div class="name">rabbit</div>,
<div class="name">fox</div>,
<div class="name">bear</div>,
<div class="name">lion</div>,
<div class="name">tiger</div>]
```

예제3. 예제2에서 긁어온 데이터에서 html코드 말고 text만 출력하시오!

```
f = open("c:₩₩data₩₩ecologicalpyramid.html")
soup = BeautifulSoup( f, "html.parser")
result = soup.find_all(class_ ="name")
for i in result: #result 리스트안의 요소를 하나씩 가져옵니다
        print(i.get_text())
```

결과:

```
plants
algae
```

deer
rabbit
fox
bear
lion
tiger

설명: get_text()함수를 사용하면 html 코드 말고 텍스트만 가져옵니다.

12/16 (142-143)

2020년 12월 16일 수요일 오전 9:46

⊙ 빅데이터 수집을 위해 반드시 알아야 할 필수 기술? 웹스크롤링
⊙ 웹스크롤링을 위해서 사용한 파이썬 모듈? beautiful soup 모듈
⊙ 웹스크롤링을 편하게 잘 하기 위한 비법?
크롬의 개발자 모드(F12)로 들어가서 화살표를 이용해서 웹상에서 스크롤링하기 원하는 지점의 html클래스 이름을 알아냄

■ 142 웹스크롤링 실전 1단계 (ebs 레이디 버그 게시판)

우리회사의 신제품이 출시 되었을 때 그 제품에 대한 사람들의 반응을 데이터 분석을 하고자 할 때 웹스크롤링 + 데이터 시각화를 이용하면 됩니다.

예제1. ebs레이디버그 시청자 게시판의 url을 가지고 직접 html문서를 내려 받을 수 있도록 코드를 구현
(어제는 우리가 ctrl + s를 누르고 웹페이지를 우리 피씨에 저장하고 구현을 했는데 오늘은 그렇게 하지 않고 웹에서 바로 html문서를 내려받을 수 있도록 구현)

```
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
print(url) #<urllib.request.Request object at 0x000002C7E03005B0>
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
print(result) #위의 url의 html전체 문서가 다 출력되고 있음.
```

결과:

```
<li><a href="http://www.ebse.co.kr/" target="_blank" title="새창"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn1.gif" alt="EBS english" /></a></li>
<li><a href="http://www.ebslang.co.kr" target="_blank" title="새창"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn2.gif" alt="EBS lang" /></a></li>
<li><a href="http://www.ebsmath.co.kr" target="_blank" title="새창"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn3.gif" alt="EBS MATH" /></a></li>
<li><a href="https://clipbank.ebs.co.kr/main?src=text&kw=00006B"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn4.gif" alt="EBS 클립뱅크" /></a></li>
<li><a href="http://www.eidf.co.kr/dbox" target="_blank" title="새창"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn5.gif" alt="EIDF DBOX" /></a></li>
<li><a href="http://ebsstory.blog.me" target="_blank" title="새창"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn6.gif" alt="EBS 스토리 기업블로그" /></a>
</li>
<li><a href="https://www.ebs.co.kr/customer/twitter"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn7.gif" alt="Twitter EBS SNS" /></a></li>
<li><a href="http://book.ebs.co.kr"><img
src="https://static.ebs.co.kr/www/common/img/category/main/footer_new_bn8.gif" alt="Book Mall" /></a></li>
</ul>
</div>
</div> -->
```

```
<div class="wrap_policy">
<div class="inner">
<div id="policy" class="policy">

<p><a href="https://about.ebs.co.kr/kor/main/index">EBS소개</a><span>·</span>  <a
href="https://about.ebs.co.kr/kor/notice/announce">EBS공고</a><span>·</span><a
href="https://home.ebs.co.kr/copyright/index.html">저작권 침해 제보</a><span>·</span><a
href="https://home.ebs.co.kr/green">EBS클린신고</a><span>·</span><a
href="https://about.ebs.co.kr/kor/business/concert"><strong>제휴문의</strong></a><span class="pct-block">·</span>
</p>
<p><a href="https://about.ebs.co.kr/kor/business/sponsorship"><strong>광고문의</strong></a><span>·</span><a
href="https://sso.ebs.co.kr/idp/policy/termsinfo/termsInfo.jsp">이용약관</a><span>·</span><a
href="https://sso.ebs.co.kr/idp/policy/privacy/privacy.jsp"  style="color: #3c73a0; font-weight: bold;">개인정보처리방침
</a><span>·</span><a href="https://sso.ebs.co.kr/idp/policy/youthinfo/youthinfo.jsp">청소년 보호정책</a></p>
</div>
 <div class="area_link">
<div class="family_link">
      <button type="button" title="FAMILY SITE" class="btn_family_link"><span class="screen_out">FAMILY
      SITE</span></button>
      <ul class="family_link_layer">
            <li><a title="새창" href="http://primary.ebs.co.kr" target="_blank">EBS 초등</a></li>
            <li><a title="새창" href="http://mid.ebs.co.kr" target="_blank">EBS 중학</a></li>
            <li><a title="새창" href="http://www.ebsi.co.kr" target="_blank">EBSi (고교)</a></li>
            <li><a title="새창" href="http://www.ebse.co.kr/ebs/index.laf" target="_blank">EBSe (영어)</a></li>
            <li><a title="새창" href="http://www.ebslang.co.kr/ebs/index.laf" target="_blank">EBS랑 (외국어) </a></li>
            <li><a title="새창" href="http://www.ebsmath.co.kr/" target="_blank">EBS MATH (수학)</a></li>
            <li><a title="새창" href="http://www.ebssw.kr/" target="_blank">이숖 (소프트웨어)</a></li>
            <li><a title="새창" href="https://www.ebs.co.kr/space/main" target="_blank">스페이스공감 (공연)</a></li>
            <li><a title="새창" href="http://www.eidf.org/kr" target="_blank">EIDF (다큐영화제)</a></li>
            <li><a title="새창" href="https://about.ebs.co.kr/kor/social/main" target="_blank">사회공헌 홈페이지</a>
            </li>
            <li><a title="새창" href="https://www.ettc.co.kr" target="_blank">EBS원격교육연수원</a></li>
            <li><a title="새창" href="https://www.ebslang.co.kr/speakingEng/main.ebs?frame_url=" target="_blank">EBS
            전화/회상 외국어</a></li>

            <li><a title="새창" href="http://ebsmedia.kr/home/" target="_blank">EBS 미디어</a></li>
            <li><a title="새창" href="http://www.ebs.co.kr/kids" target="_blank">EBS KIDS</a></li>
      </ul>
</div>
<a href="http://0302.co.kr" target="_blank" class="pct-block"><img src="/common/images/global/btn_remote.png"
alt="원격지원"></a>
<a href="https://www.ebs.co.kr/customer/question/faq" target="_blank" class="pct-block"><img
src="/common/images/global/btn_faq.png" alt="FAQ"></a>
<div class="call_help">
      <a href="mailto:helpdesk@ebs.co.kr">
            <img src="/common/images/global/ico_tell.png" alt="1588-1580 helpdesk@ebs.co.kr">
            <p>1588-1580<span class="pct-block">helpdesk@ebs.co.kr</span></p>
      </a>
</div>
</div>
<!-- 2020-05-19 -->
<div class="edu_confirm">
 <img src="/common/images/global/img_footer_pims_2020.png" alt="" />
 <p><span>[인증범위] 교육포털 웹서비스</span><span>[유효기간] 2020.04.29 ~ 2023.04.28</span></p>
</div>
<!-- // 2020-05-19 -->
<div class="area_author">
<address>
      <strong>10393</strong> 경기도 고양시 일산동구 한류월드로 281 한국교육방송공사 <span>사업자등록번호 :
      229-82-01343 (한국교육방송공사사장 김명중)</span>
      <p>시청자불만전담 : <a href="hotline@ebs.co.kr" target="_blank">hotline@ebs.co.kr</a><span>부가통신사업신
```

```
고 : 10077  통신판매업신고 : 고양일산동-1415호  <a href="http://www.ftc.go.kr/bizCommPop.do?wrkr_no=
2298201343" target="_blank">[사업자정보확인]</a></span></p>
<strong class="copyright">Copyright ⓒ EBS. All Rights Reserved.</strong>
</address>
</div>
</div>
</div>
</div>



<!-- 퀵메뉴 영역 -->
<script type="text/javascript">
$(window).scroll(function(){
var s = $(window).scrollTop(),f = $('.footer').height(),wH = $(window).height(),f_top = $('.footer').offset();
if(s > (f_top.top-(wH+60)) ){
$('.mobQuick').addClass('posAsol');
$('.mobQuick').css('bottom',(f+10));
} else{
$('.mobQuick').removeClass('posAsol');
$('.mobQuick').css('bottom','10px');
}

});
$(document).ready(function() {
$(".rid_menu").click(function(){
var t = $(this).children('a');
var sell = $(this).siblings('.Qmenu');
if ($(sell).is(":visible") == true) {
$(sell).slideUp(200);
$(t).text('+');
$(this).removeClass('on');
} else {
$(sell).slideDown(200);
$(t).text('-');
$(this).addClass('on');
}
});
$('body').bind('touchstart', function(e) {
if($(".Qmenu").css("display") == "block"){
if(!$('.rid_menu, .Qmenu').has(e.target).length) {
$(".Qmenu").hide();
$('.rid_menu').removeClass('on').children('a').text('+');
}
}
//e.preventDefault();
});

    // 공지사항
    $.ajax({
        url:"/ladybug/function/getQuickMnuLists.json",
        success: function(data) {
            var result;
            var quickMnuCnt = data.quickMnuLists.length;
            if (quickMnuCnt > 0) {
                var listHtml = "";
                for (var i = 0, n = data.quickMnuLists.length; i < n; i++) {
                    var result      = data.quickMnuLists[i];

                    listHtml += '<li><a href="/'+result.hmpId+'/'+result.hmpMnuSno+'/hmpLink">'+result.mnuNm+'</a>
</li>';
```

```
                }
                //alert(listHtml);
                 $('.Qmenu').html(listHtml);
            } else {
                  $(".rid_menu").css("display","none");
            }
        },
        error: function(e) {
            ;
        }
    });

});
</script>

<div class="mobQuick mob-block">
<div class="rid_menu"><a href="javascript:void(0);">+</a></div>
<p class="triangle"></p>
<ul class="Qmenu">
</ul>
</div>

<!-- //퀵메뉴 영역 -->


<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$(".btn_family_link").click(function(){
if($(".family_link_layer").css("display") == "none"){
$(".family_link_layer").show();
} else {
$(".family_link_layer").hide();
}
});
});
function tab1() {
window.open("https://sso.ebs.co.kr/idp/popup/termsInfo.jsp?tab=1", "tab1", "scrollbars=no, resizeable = no, width=700,
height=670");
}
function tab2() {
if (isMobileBrowser()) {
location.href="https://m.ebs.co.kr/cs/privacy";
}else{
window.open("https://sso.ebs.co.kr/idp/popup/termsInfo.jsp?tab=2", "tab2", "scrollbars=no, resizeable = no, width=700,
height=670");
}
}
</script>

<!-- 방문자 로그 수 증가 -->
<script type="text/javascript">
$(document).ready(function() {
$.ajax({
        url: "/ladybug/ajax/ajaxVisitLogPlus.json",
        dataType : 'JSON',
        type: "get",
        cache: false,
        success: function(data) {
            return false;
        }
```

```
    });
});
function changeMobSession() {
$.ajax({
        url: "/ladybug/ajax/mobileSession.json?viewType=mob",
        dataType : 'JSON',
        type: "get",
        success: function(data) {
            //return false;
          document.location.href="https://m.ebs.co.kr/ladybug";
        }
    });
}
</script>

<!-- // 공통 푸터 -->
<script type="text/javascript">
function win_popup(url) {
window.open(url, "_blank", "width=700,height=670,scrollbars=no,location=no");
}
</script>



<!-- AceCouter -->




<!-- This script is for AceCounter START -->
<script>

</script>
<!-- AceCounter END -->

<!-- AceCounter Log Gathering Script V.71.2010011401 -->

<!-- AceCounter Log Gathering Script End -->

<script type="text/javascript">
function removeOldCookie(name,domain,path){
var d=new Date();
d.setTime(d.getTime()-1);
document.cookie = name + "=; path="+path+"; doamin="+domain+"; expires="+d.toGMTString();
}

removeOldCookie("GsPID_1","wlog.ebs.co.kr","/");
removeOldCookie("GsPID_2","wlog.ebs.co.kr","/");
removeOldCookie("GsPID_3","wlog.ebs.co.kr","/");
removeOldCookie("GsPID_4","wlog.ebs.co.kr","/");
removeOldCookie("GsPID_5","wlog.ebs.co.kr","/");
</script>

<!-- //AceCouter -->

<!-- nethru -->

<!-- WISELOG ONLINE TRACKING SCRIPT CODE START -->
<!-- DO NOT MODIFY THIS SCRIPT CODE. -->
<!-- COPYRIGHT (C) 1999-2013 NETHRU INC. ALL RIGHTS RESERVED. -->
<script type="text/javascript">
```

```
(function() {
var nl = document.createElement('script'); nl.type = 'text/javascript';
nl.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'home.ebs.co.kr/common/js/wlo.js';
var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(nl, s);
var done = false;
nl.onload = nl.onreadystatechange = function() {
if ( !done && (!this.readyState || this.readyState === "loaded" || this.readyState === "complete") ) {
done = true;
_n_sid = "home.ebs.co.kr";
if (hmpId == 'jisike' || hmpId == 'bestdoctors' || hmpId == 'docuprime' || hmpId == 'worldtrip'){
_n_sid = hmpId+".ebs.co.kr";
}
_n_uid_cookie = "DS_CIF";
n_logging();
nl.onload = nl.onreadystatechange = null;
}}})();
</script>
<!-- WISELOG ONLINE TRACKING SCRIPT CODE END -->
<!-- //nethru -->

<!-- 구글 스크립트 추가 2015.12.09 mjseo -->
<script>



  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');
  ga('create', 'UA-70996157-2', 'auto');
  ga('create', 'UA-170906430-1', 'auto', 'clientTracker');
  ga('send', 'pageview');
  ga('clientTracker.send','pageview');




</script>
<!-- 구글 스크립트 추가 끝 -->

<!-- 광고 관련 sns tracker 추가 건 -->



<script>
  ga('create', 'UA-96786084-2', 'auto', 'snsTracker');
  ga('snsTracker.send', 'pageview');
</script>


<script>
!function(f,b,e,v,n,t,s){if(f.fbq)return;n=f.fbq=function(){n.callMethod?
n.callMethod.apply(n,arguments):n.queue.push(arguments)};if(!f._fbq)f._fbq=n;
n.push=n;n.loaded=!0;n.version='2.0';n.queue=[];t=b.createElement(e);t.async=!0;
t.src=v;s=b.getElementsByTagName(e)[0];s.parentNode.insertBefore(t,s)}(window,
document,'script','https://connect.facebook.net/en_US/fbevents.js');
fbq('init', '417758951909111'); // Insert your pixel ID here.
fbq('track', 'PageView');
</script>
```

```
<noscript><img height="1" width="1" style="display:none"
src="https://www.facebook.com/tr?id=417758951909111&ev=PageView&noscript=1"
/></noscript>
<!-- DO NOT MODIFY -->



<script type="text/javascript">
var wptgStr = "web";
var criteoStr = "d";
var snsTrackerMobBrowserYn = "N";

if (snsTrackerMobBrowserYn=="Y") {
    var roosevelt_params = {
            retargeting_id:'VcdOBD3OyPFKF-s2E5lirA00',
            tag_label:'nZRoeVbVRdCOMH2niBj9FQ'
        };
    wptgStr = "mobile";
    criteoStr = "m";
} else {
    var roosevelt_params = {
            retargeting_id:'VcdOBD3OyPFKF-s2E5lirA00',
            tag_label:'-S58NUJPQPerDF660CxorA'
        };
}
</script>
<script type="text/javascript" src="//adimg.daumcdn.net/rt/roosevelt.js" async></script>


<!-- ##사이트 공통 -->


<!-- ##사이트 공통 -->

<script type="text/javascript" src="//static.criteo.net/js/ld/ld.js" async="true"></script>
<script type="text/javascript">
window.criteo_q = window.criteo_q || [];
window.criteo_q.push(
        { event: "setAccount", account: 9858 }, /* 고정값 */
        { event: "setSiteType", type: criteoStr }, /* "m" 모바일&테블릿 페이지는 "m" 데스크탑 페이지는 "d" */
        { event: "viewHome", visit_page: "0" } /* 고정값 */
);
</script>

<!--  // ---->



<!-- 2020-11-02 트래킹,트레킹 소스 삽입 -->
<!-- ADN Tracker[공통] start -->
<script type="text/javascript">
var adn_param = adn_param || [];
adn_param.push([{
    ui:'102609',
    ut:'Home'
}]);
</script>
<script type="text/javascript" async src="//fin.rainbownine.net/js/adn_tags_1.0.0.js"></script>
<!-- ADN Tracker[공통] end -->

<!-- ADN 클로징패널 설치  start -->
```

```
<!-- ADN 클로징패널 설치  start -->
<script type="text/javascript">
var adn_panel_param = adn_panel_param || [];
adn_panel_param.push([{
 ui:'102609',
 ci:'1026090001',
 gi:'32407'
}]);
</script>


<script type="text/javascript" async src="//fin.rainbownine.net/js/adn_closingad_1.1.1.js"></script>
<!-- ADN 클로징패널 설치 end -->

<!-- ADN 크로징 설치  start -->
<script type="text/javascript">
var adn_mobile_panel_param = adn_mobile_panel_param || [];
adn_mobile_panel_param.push([{
ui:'102609',
ci:'1026090002',
gi:'32408'
}]);
</script>
<script type="text/javascript" async src="//fin.rainbownine.net/js/adn_mobile_closingad_1.1.2.js"></script>
<!-- ADN 크로징 설치 end -->



<!-- 공통 적용 스크립트 , 모든 페이지에 노출되도록 설치. 단 전환페이지 설정값보다 항상 하단에 위치해야함 -->
<script type="text/javascript" src="//wcs.naver.net/wcslog.js"> </script>
<script type="text/javascript">
if (!wcs_add) var wcs_add={};
wcs_add["wa"] = "s_24868b5c33e6";
if (!_nasa) var _nasa={};
wcs.inflow();
wcs_do(_nasa);
</script>


<!-- //광고 관련 sns tracker 추가 건 -->

<script type="text/javascript">
BHP.UI.Layout.init('009');
var layoutCd = '009';
BHP.UI.GnbLnb.init('009');
if ($("#contents .admin_content").length) {
$(".page_grid3").removeClass("page_grid3");
$("#contents, .bhp").attr("style", "float:none; display:block; width:770px !important; margin:0 auto; padding:0 100px 30px;
background:#fff;");
$(".custom_area").next().remove();
$(".snb, .aside, .custom_area, .spot_info").remove();
}
</script>
```

예제2. 예제1에서 긁어온 html코드를 Beautiful soup 의 함수를 이용해서 웹스크롤링을 할 수 있도록 Beautiful soup을 쓸 수 있게 파싱하시오.

```
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")
print(soup)
```

예제3. 지금 페이지의 시청자 게시판의 글 내용에 해당하는 부분의 테그와 클래스 이름을 알아내시오.

```
< p class_con="con"> 재미있 다 < /p>
```

예제4. p테그중에 class가 con에 해당하는 부분을 스크롤링 하시오!

1.

앞의 코드들...

```
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")
result2 = soup.find_all( 'p', class_ ='con')
print(result2)
```

2.

앞의 코드들...

```
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")
result2 = soup.find_all( 'p', class_ ='con')
print(result2)
```

설명 : find함수는 맨 처음 하나만 가져오는데 find_all은 p테그에 class이름 con에 해당하는 부분을 모두 가져온다.

예제5. 위의 결과에서 html문서 말고 한글 텍스트만 가져오시오!

(위의 result2는 리스트 입니다)

앞의 코드들...

```
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")
result2 = soup.find_all( 'p', class_ ='con')
for i in result2:
    print(i.get_text())
```

예제6. 예제5에서 출력되고 있는 텍스트들이 좀더 깔끔하게 나오게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
```

```
soup = BeautifulSoup( result, "html.parser")
result2 = soup.find_all( 'p', class_ ='con')
for i in result2:
    print(i.get_text(" ", strip=True))
```

예제7. 예제6에서 출력되고 있는 텍스트들이 params라는 비어있는 리스트에 담기게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")
result2 = soup.find_all( 'p', class_ ='con')
params=[]
for i in result2:
    params.append(i.get_text(" ", strip=True))
print(params)
```

결과:

['재미있 다', '어제 흐름상 종영 같지 않았는데 갑자기 끝이네요??', '〈미라큘러스: 레이디버그와 블랙캣〉종영 안내!₩r₩n₩r₩n₩r₩n₩r₩n〈미라큘러스: 레이디버그와 블랙캣〉-시즌3가 ₩r₩n₩r₩n방송 종료되었습니다, 그동안 시청해주셔서 감사드립니다!₩r₩n₩r₩n2021년 새로운 시즌으로 여러분을 찾아뵙도록 노력하겠습니다!₩r₩n₩r₩n고맙습니다~', '아이들 때문에 보게 되었는데 어른인 제가 더 좋아하는 프로그램이 되었네요 아이들이랑 볼 때 너무 신나요 함께 볼 수 있는 보기 드문 애니매이션이라 더 좋네요 시즌 4도 너무 너무 기대되고 기다리고 있습니다', '2021년이 빨리되서 레이디버그 시즌4를 보면 좋겠네요~', '레이디버그... 제가 어렸을때도 너~~무 좋아 하던 방송 이에요!!!!!!!', '이렇게 재미있는프로는 이 세상에 하나뿐이예요 아 그리고 뉴욕편은 언제 올려주시나요?', '너모 재밌어요!!! 저 시즌 1부터 다 본방으로 챙겨본 미라큘러 입니다ㅋㅋㅋ 진짜 재밌구요 팬카페랑 이런데도 다 가입해있어요... 방도 레이디버그 사진으로 덕-질☆', '완전 재밌다', '근뎅.....오즈 같나서....레이디버그가 하는거잖아여....ㅠ', '우왕!', '넘재밌어요.매일하면 좋겠어요', '재밌어요 완전 꿀잼이요 사랑합니다', 'ㅎㅎ', '진짜여???', '보니하니,레이디버그,형사 가제트,오즈,삐삐에 있었어요!', '오늘부터 미라큘러스 시즌3 다시 방영한다네요!^^₩r₩n목,금 저녁 7시', '전 세매큐,픽시,가제트형사,오:마법을 찾아서,히어로써클,로빈후드에 있어여!ㅎㅎ즈', '벌써 가을 인길요~', '가을이 빨리 되면 좋겠어요']

예제8. 게시글을 올린 날짜를 스크롤링하기 위해 게시글 날짜가 있는 html문서의 테그 이름과 클래스 이름을 확인하시오!

F12 개발자모드를 사용해서 알아보기!

테그이름은 span이고 클래스 이름은 date입니다.

예제9. 예제8의 코드에 날짜를 스크롤링하는 코드를 추가하고 위의 날짜를 모두 스크롤링해서 params2라는 리스트에 담으시오!

```
# 1. 웹에서 html문서를 가져와 beautifulsoup으로 파싱
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

# 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
result1 = soup.find_all( 'p', class_ ='con')
result2 = soup.find_all( 'span', class_ ='date')

#시청자 게시판의 날짜와 본문을 params와 params2에 담습니다.
params1=[]
params2=[]
for i in result1:
```

```
    params1.append(i.get_text(" ", strip=True))
for i in result2:
    params2.append(i.get_text(" ", strip=True))
print(params1)
print(params2)
```

결과:
['재미있 다', '어제 흐름상 종영 같지 않았는데 갑자기 끝이네요??', '〈미라큘러스: 레이디버그와 블랙캣〉종영 안내!₩r₩n₩r₩n₩r₩n〈미라큘러스: 레이디버그와 블랙캣〉-시즌3가 ₩r₩n₩r₩n방송 종료되었습니다, 그동안 시청해주셔서 감사드립니다!₩r₩n₩r₩n2021년 새로운 시즌으로 여러분을 찾아뵙도록 노력하겠습니다!₩r₩n₩r₩n고맙습니다~', '아이들 때문에 보게 되었는데 어른인 제가 더 좋아하는 프로그램이 되었네요 아이들이랑 볼 때 너무 신나요 함께 볼 수 있는 보기 드문 애니매이션이라 더 좋네요 시즌 4도 너무 너무 기대되고 기다리고 있습니다', '2021년이 빨리되서 레이디버그 시즌4를 보면 좋겠네요~', '레이디버그... 제가 어렸을때도 너~~무 좋아 하던 방송 이에요!!!!!!!', '이렇게 재미있는프로는 이 세상에 하나뿐이예요 아 그리고 뉴욕편은 언제 올려주시나요?', '너모 재밌어요!!! 저 시즌 1부터 다 본방으로 챙겨본 미라큘러 입니다ㅋㅋㅋ 진짜 재밌구요 팬카페랑 이런데도 다 가입해있어요... 방도 레이디버그 사진으로 덕-질☆', '완전 재밌다', '근뎅.....오즈 같나서....레이디버그가 하는거잖아여....ㅠ', '우왕!', '넘재밌어요.매일하면 좋겠어요', '재밌어요 완전 꿀잼이요 사랑합니다', 'ㅎㅎ', '진짜여???', '보니하니,레이디버그,형사 가제트,오즈,삐삐에 있었어요!', '오늘부터 미라큘러스 시즌3 다시 방영한다네요!^^₩r₩n목,금 저녁 7시', '전 세매큐,픽시,가제트형사,오:마법을 찾아서,히어로써클,로빈후드에 있어여!ㅎㅎ즈', '벌써 가을 인길요~', '가을이 빨리 되면 좋겠어요']

['2020.12.11 19:52', '2020.12.11 19:06', '2020.12.11 01:35', '2020.12.10 12:43', '2020.10.17 15:30', '2020.10.16 12:25', '2020.10.15 21:04', '2020.09.29 14:46', '2020.09.29 11:30', '2020.09.21 08:17', '2020.09.21 08:16', '2020.09.20 09:07', '2020.09.19 22:52', '2020.09.16 12:34', '2020.09.14 10:26', '2020.09.11 09:45', '2020.09.11 09:42', '2020.09.06 08:58', '2020.09.05 21:19', '2020.09.04 18:14']

예제10. 위의 날짜와 본문 내용이 아래와 같이 출력되게 하시오!
2020.12.11 19:52 재미있 다
2020.12.11 19:06 어제 흐름상 종영 같지 않았는데 갑자기 끝이네요??
    :   :

답:
```
# 1. 웹에서 html문서를 가져와 beautifulsoup으로 파싱
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

# 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
result1 = soup.find_all( 'span', class_ ='date')
result2 = soup.find_all( 'p', class_ ='con')

#시청자 게시판의 날짜와 본문을 params와 params2에 담습니다.
params1=[]
params2=[]
for i in result1:
    params1.append(i.get_text(" ", strip=True))
for i in result2:
    params2.append(i.get_text(" ", strip=True))


# 4. 날짜와 본문을 같이 출력 합니다.
for k, h in zip(params1, params2):
    print(k+' ' + h)
```

예제11. (중요) 레이디 버그 게시판 전체의 글을 다 스크롤링해서 예제 10번 결과처럼 출력되게하시오!

페이지1
http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&
페이지2
http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&
페이지3
http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=3&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&

:
위의 url을 가만히 살펴보면 페이지 번호만 다르고 나머지는 다 같다는 것을 확인할 수 있습니다.

```
# 1. 웹에서 html문서를 가져와 beautifulsoup으로 파싱
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuId=106
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

# 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
result1 = soup.find_all( 'span', class_ ='date')
result2 = soup.find_all( 'p', class_ ='con')

#시청자 게시판의 날짜와 본문을 params와 params2에 담습니다.
params1=[]
params2=[]
for i in result1:
    params1.append(i.get_text(" ", strip=True))
for i in result2:
    params2.append(i.get_text(" ", strip=True))

# 4. 날짜와 본문을 같이 출력 합니다.
for k, h in zip(params1, params2):
    print(k+' ' + h)


#노가다하기
```

예제12. 아래의 결과를 출력하는데 페이지 번호가 1~22번까지 변하게 하시오.
http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&

http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&

http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=3&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&

```
for i in range(1,23):
    print('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106
    &searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&') #앞
    에도 문자고 뒤에도 문자라 문자끼리만 결합 가능해 문자로 감싸줘야함
```

예제13. 예제12번 코드를 예제10번 코드에 적용해서 레이디 버그 전체 게시판의 글들이 날짜와 함께 출력되게 하시오!

1. **예제 10번 코드를 가져옵니다**

```
# 1. 웹에서 html문서를 가져와 beautifulsoup으로 파싱
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

# 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
result1 = soup.find_all( 'span', class_ ='date')
result2 = soup.find_all( 'p', class_ ='con')

#시청자 게시판의 날짜와 본문을 params와 params2에 담습니다.
params1=[]
params2=[]
for i in result1:
    params1.append(i.get_text(" ", strip=True))
for i in result2:
    params2.append(i.get_text(" ", strip=True))

# 4. 날짜와 본문을 같이 출력 합니다.
for k, h in zip(params1, params2):
    print(k+' ' + h)
```

2. **예제10번 코드에서 url의 페이지 번호가 1번부터 22번까지 변경되면서 날짜와 게시글을 가져올 수 있도록 코드를 수정하시오!**

```
for i in range(1,23):
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
#%%
# 1. 웹에서 html문서를 가져와 beautifulsoup으로 파싱
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

for i in range(1,23):
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    # 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
    result1 = soup.find_all( 'span', class_ ='date')
    result2 = soup.find_all( 'p', class_ ='con')

    #시청자 게시판의 날짜와 본문을 params와 params2에 담습니다.
    params1=[]
    params2=[]
    for i in result1:
        params1.append(i.get_text(" ", strip=True))
    for i in result2:
        params2.append(i.get_text(" ", strip=True))
```

```
    # 4. 날짜와 본문을 같이 출력 합니다.
    for k, h in zip(params1, params2):
        print(k+' ' + h)
```

설명 : 위의 params1과 params2가 리스트 for문 안쪽에 있기 때문에 페이지 번호가 돌때마다 params 리스트 안의 내용이 새로운 날짜와 내용으로 변경되게 됩니다.

예제14. params1과 params2에 레이디 버그 시청자 게시판의 모든 게시날짜와 본문 내용이 들어가게 코드를 수정하시오!

```
for i in range(1,23): #페이지 번호를 1번부터 22번까지 변경하려고 for문을 사용
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'

# 0. 웹스크롤링에 필요한 모듈을 임포트 합니다.
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 1. 레이디 버그 게시판 게시날짜와 게시글 전체를 다 담은 리스트 2개를 만듭니다.
params1=[] #게시날짜를 담을 것이고
params2=[] #게시본문을 담을 것 입니다.

for i in range(1,23):
        # 2. 웹에서 html 문서를 가져와 beautifulsoup으로 파싱
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    # 3. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
    result1 = soup.find_all( 'span', class_ ='date')
    result2 = soup.find_all( 'p', class_ ='con')

    # 4. 시청자 게시판의 날짜와 본문을 params와 params2 리스트에 담습니다.
    #아까는 이자리에 params1=[], params2 = []가 있어서 페이지번호가 변경될때마다
    # params 리스트가 초기화가 되었었는데 지금은 이 자리에 없고 맨 위에 for문 시작하기 전에 있었으므로
    # 계속 페이지번호 1~ 페이지번호 22번까지의 모든 게시날짜와 게시본문이 params1과 params2에 append 됩니다.
    for i in result1:
        params1.append(i.get_text(" ", strip=True))
    for i in result2:
        params2.append(i.get_text(" ", strip=True))

# 5. 날짜와 본문을 같이 출력 합니다.
for k, h in zip(params1, params2):
    print(k+' ' + h)
```

설명: 예제13번은 레이디버그 게시판 전체글을 화면에 출력하는 코드이고 예제14는 레이디버그 게시판 전체글을 params1과 params2 리스트에 모두 담고 화면에 출력하는 코드

■ 143. 웹스크롤링 실전2 (중앙일보사)

중앙일보사 홈페이지에서 인공지능으로 검색을 했을 때 나오는 기사들을 전부 웹스크롤링 한다.

예제1. 중앙일보사 홈페이지에서 인공지능으로 검색했을 때 나오는 url을 가져오시오.
https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews

예제2. 위의 사이트에서 보이는 상세기사를 클릭하고 그 기사의 url을 복사하시오!
https://news.joins.com/article/23947044
https://news.joins.com/article/23946979
https://news.joins.com/article/23946876

예제3. 인공지능으로 검색했을 때 나오는 상세기사들의 url을 스크롤링 하시오!

#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

#2. 중앙일보에서 인공지능으로 검색했을 때 나오는 첫 페이지의 html코드를
# beautiful soup에서 이용할 수 있도록 파싱합니다.
list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup(result, "html.parser")

#3. 상세기사 url을 가져올 수 있도록 테그와 클레스를 찾습니다.
#찾아보니 테그는 h2이고 클래스 이름은 headline mg입니다.

result1 = soup.find_all('h2', class_ = 'headline mg')
print(result1)

#4. 위의 result1은 리스트 이므로 for loop문을 이용해서 리스트에 있는 요소를 하나씩 빼냅니다.
#하나씩 빼내면서 href의 값을 얻어냅니다.

여기서
```
for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    print(i)
```

이렇게 하고 보면:
<h2 class="headline mg">파주~대전 500㎞ 자율주행 트럭, 박사 한명 없이 만든 韓벤처</h2>
이렇게 결과가 나온다.

이다음에

```
for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    print(i.get('href'))
```

이렇게 하면 결과가 안나온다.

따라서

```
for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
        print(k)
```

이렇게 해주면 h2안의 a테그에 관한 요소가 다 나온다.

```
<a href="https://news.joins.com/article/23947044" target="_blank">'AI 발전에 써라' 동원 김재철 명예회장, KAIST에 500억 기부</a>
```

이렇게..

따라서 여기서

```
for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
        print(k.get('href'))
```

이렇게 해주면

https://news.joins.com/article/23947044
https://news.joins.com/article/23946979
https://news.joins.com/article/23946876
https://news.joins.com/article/23946841
https://news.joins.com/article/23946757
https://news.joins.com/article/23946642
https://news.joins.com/article/23946268
https://news.joins.com/article/23946216
https://news.joins.com/article/23946191
https://news.joins.com/article/23946032

이렇게 url만 나온다.

예제4. 위의 상세 기사 url을 params라는 비어있는 리스트에 다 append 시키시오.

```
#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

#2. 중앙일보에서 인공지능으로 검색했을 때 나오는 첫 페이지의 html코드를
#   beautiful soup에서 이용할 수 있도록 파싱합니다.
list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

#3. 상세기사 url을 가져올 수 있도록 테그와 클레스를 찾습니다.
#찾아보니 테그는 h2이고 클래스 이름은 headline mg입니다.

result1 = soup.find_all( 'h2', class_ = 'headline mg')

#4. 위의 result1은 리스트 이므로 for loop문을 이용해서 리스트에 있는 요소를 하나씩 빼냅니다.
#하나씩 빼내면서 href의 값을 얻어냅니다.

params = []

for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
        params.append(k.get('href'))
```

```
print(params)
```

예제5. 상세기사 url중에 하나를 복사해 오고 그 상세기사 url의 웹페이지로 접속해서 본문기사의 테그 이름과 클래스 이름이 무엇인지 확인하시오!

https://news.joins.com/article/23947044
답: 테그이름은 div, z클래스 이름은 'article_body mg fs4'

예제6. 위의 상세기사의 본문 텍스트를 스크롤링해서 출력하시오!

```
#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

#2. 상세기사 url로 검색햇을 때 나오는 ᄑᆞᆼ ᅦ이지의 html코드를
#beautiful soup에서 이용할 수 있도록 파싱합니다.
list_url = 'https://news.joins.com/article/23947044'
url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
soup = BeautifulSoup( result, "html.parser")

#3. 상세기사 url을 가져올 수 있도록 테그와 클레스를 찾습니다.
#찾아보니 테그는 div이고 클래스 이름은 article_body mg fs4입니다.

result1 = soup.find_all( 'div', class_ = 'article_body mg fs4')

#4. 위의 result1은 리스트 이므로 for loop문을 이용해서 리스트에 있는 요소를 하나씩 빼냅니다.
#하나씩 빼내면서 href의 값을 얻어냅니다.

params2 = []
for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
    params2.append( i.get_text(" ", strip=True) )

print(params2)
```

예제7. 상세기사 url을 가져와서 params 리스트에 넣고 출력하는 예제4번 코드를 함수로 생성하시오!
print(j_scroll())

```
from bs4 import BeautifulSoup
import urllib.request

def j_scroll():

    list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")


    result1 = soup.find_all( 'h2', class_ = 'headline mg')

    params = []

    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
            params.append(k.get('href'))
```

```
    return params

print(j_scroll())
```

예제8. 상세 기사 url 로 본문 기사를 스크롤링해서 리스트에 담았던 예제6번을 j_detail_scroll() 함수로 생성하시오 !

```
#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
from bs4 import BeautifulSoup
import urllib.request


def j_detail_scroll():
    list_url = 'https://news.joins.com/article/23947044'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    result1 = soup.find_all( 'div', class_ = 'article_body mg fs4')

    params2 = []
    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        params2.append( i.get_text(" ", strip=True) )

    return(params2)

print(j_detail_scroll())
```

예제9. 지금만든 j_detail_scroll()함수는 상세기사 딱 한개의 본문을 출력하는 함수인데 이 j_detail_scroll() 함수에 j_scroll() 함수를 실행했을 때 나오는 상세 url 여러개를 제공할 수 있도록 코드를 수정하시오!

```
#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
from bs4 import BeautifulSoup
import urllib.request

def j_scroll():
    list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    result1 = soup.find_all( 'h2', class_ = 'headline mg')

    params = []

    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
            params.append(k.get('href'))

    return params


def j_detail_scroll():
    list_url = j_scroll()
    params2 = []
    for i in list_url:
        url = urllib.request.Request(i) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
        result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
        soup = BeautifulSoup( result, "html.parser")
```

```
        result1 = soup.find_all( 'div', class_ = 'article_body mg fs4')


        for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )

    return params2

print(j_detail_scroll())
```

■ 중앙일보의 기사를 스크롤링하는 내용 총정리

1. 검색키워드(예: 인공지능)를 가지고 검색을 한 후 그 url을 얻어낸다.
2. 상세기사 url을 리스트에 담는 j_scroll()함수를 생성한다.
3. 상세기사 url로 기사본문을 스크롤링하는 j_detail_scroll() 함수를 생성한다.
4. j_detail_scroll() 함수 안에 j_scroll() 함수를 호출해서 상세기사 url 여러개를 받아오게끔 코딩하고 받아온 상세기사 url로 본문 기사들을 params2에 담게끔 코드를 수정한다.

12/17

2020년 12월 17일 목요일 오전 9:44

- 어제 웹 스크롤링 한 사이트

1. 레이디버그 게시판
2. 중앙일보
3. 동아일보
4. 한겨레
5. 각자 원하는 사이트

공통적으로 웹스크롤링 안되었던 이유가 있었다.

Beautoful soup 문법 중에 좀 더 알아야 할 내용

SQL vs 파이썬

select 파이썬 자료형 5가지를 가지고 관련된 함수, 매일매일 다른 함수들 소개

from ↓

where 앞에서 배웠던 파이썬 함수들을 처음부터 다 암기할 수 없고 반복해서 쓰다보면 자연스럽게

암기가 됩니다.

↓

□ 파이썬 코드의 큰 틀을 암기하면 된다.

1. 필요한 모듈 임폴트
2. 데이터 발생 또는 수집 (예: 주사위를 던지거나 웹스크롤링을 해서 데이터 수집)
3. 수집된 데이터를 리스트에 append시켜서 저장한다.
4. 리스트에 저장된 데이터를 가지고 통계분석을 한다.
5. 결과를 출력한다.

데이터 분석가가 갖추어야할 필수 기술?

1. 웹스크롤링 기술
2. 통계로 데이터 분석
3. 머신러닝으로 데이터 분석
4. 딥러닝으로 데이터 분석

■ beautiful soup 기능으로 웹스크롤링할 때 추가로 알아야 할 사항

예제: 웹스크롤링을 할 사이트에 접속합니다.
https://futurechosun.com/?s=%EB%B4%89%EC%82%AC

예제2. 위의 사이트에서 키워드 '봉사'로 검색을 합니다.
https://futurechosun.com/?s=%EB%B4%89%EC%82%AC
https://futurechosun.com/page/2?s=%EB%B4%89%EC%82%AC
https://futurechosun.com/page/3?s=%EB%B4%89%EC%82%AC

예제3. 위의 첫페이지의 html코드를 파이썬으로 불러오는 작업

#1. 웹스크롤링에 필요한 모듈을 임폴트 합니다.
```
from bs4 import BeautifulSoup
import urllib.request     # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈
```

#2. 첫페이지의 url을 파싱합니다. 파싱: 파이썬이 알아볼 수 있는 기계어로 바꿔준다.
```
list_url = 'https://futurechosun.com/page/1?s=%EB%B4%89%EC%82%AC'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
print(result)
```

#3. 웹스크롤링 점문 모듈인 beautiful soup 모듈을 이용할 수 있게끔 내려받은 html코드를
beautiful soup을 이용할 수 있게 파싱합니다. 이용할 수 있게 한다는 것은 beautiful soup의
함수인 find_all을 이용해서 테그 이름과 클래스 이름을 가지고 검색하기 원하는 지점을 빠르게
찾아가기 위해서 파싱한다는 것입니다.
앞의 코드들..
```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result,'html.parser')
print(soup)
```

#4. 더 나은 미래 첫페이지에 보이는 기사들 12개의 url을 알아내기 위해 테그 이름과 클래스 이름이 무엇인지
알아내시오~ (href가 있는 곳을 찾는다)

답: 테그이름은 div이고 클래스 이름은 elementor-post__title 입니다.

※ 중요 설명: href와 가장 가까이 있는 테그 이름과 클래스 이름을 알아내야 합니다.

#5. 테그이름 div에 클래스 이름 elementor-post__title에 해당하는 beautiful soup으로 파싱된 html 코드들을 모두 긁어오시오~

```
#앞의 코드들...
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
print(result1)
```

결과:

```
<div class="elementor-post__title">
<a href="https://futurechosun.com/archives/52491">
"편견에 주눅 들었던 결혼 이주 여성들... '봉사'로 자존감 되찾았다죠" </a></div>,

<div class="elementor-post__title">
<a href="https://futurechosun.com/archives/47330">
[글로벌 이슈] 전 세계 코로나19 확산으로... 해외 봉사 '올스톱' 위기 </a></div>
```

설명:

지금 위의 result1의 결과를 보면 리스트로 되어져있고 리스트 안의 각 요소들이 div테그가 위 아래로 있고 안쪽에 a 테그가 있는 구조로 beautifulsoup으로 파싱된 html코드들이 요소로 구성되어 있습니다.

우리가 원하는 지점은 a 테그인데 a 테그로 접근하려면 먼저 result1 리스트의 요소들을 하나씩 빼내와야 합니다. 리스트 안의 요소를 하나씩 빼내올려면 뭘 써야 하나요??

----> for loop문을 사용해야 합니다.

#6. result1 리스트의 요소들(beautiful soup으로 파싱된 html코드들)을 for loop문으로 하나씩 빼냅니다.

```
#앞의 코드들..
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
for i in result1:
        print(i)
```

#7. 빼낸 요소인 위의 html 코드 3줄에서 a 테그에 해당하는 부분만 가져오게 하시오.

```
#앞의 코드들
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
for i in result1:
        print(i.find_all('a'))
```

```
# 8. 위의 a테그의 html코드들 담은 리스트는 요소를 딱 하나만 담고 있다.
#    그래서 리스트안의 그 요소만 딱 뽑아내서 출력을 하시오!
```

힌트:

```
a = [2]
```

```
print(a[0]) #2
```

따라서

```
#앞의 코드들
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
for i in result1:
        print(i.find_all('a')[0])
```

#9. 그러면 뽑아낸 a 테그의 html문서에서 href의 값만 얻어내시오~

```
#앞의 코드들
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
for i in result1:
        print(i.find_all('a')[0].get("href"))
```

#10. 위의 상세기사 url을 params라는 리스트에 append 시킨다.

```
soup = BeautifulSoup( result,'html.parser')
result1=soup.find_all('div',class_ ='elementor-post__title')
params = []
for i in result1:
        params.append(i.find_all('a')[0].get("href"))

print(params)
print(len(params)) #24
```

#11. 첫페이지만 가져오는게 아니라 1페이지부터 3페이지까지 24x3 = 72 페이지의 상세 기사 url을 params 리스트에 append 시키시오.

```
from  bs4  import  BeautifulSoup
import  urllib.request   # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

params = []
for i in range(1,4):
    list_url = 'https://futurechosun.com/page/'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result,'html.parser')
    result1=soup.find_all('div',class_ ='elementor-post__title')
    for i in result1:
        params.append(i.find_all('a')[0].get("href"))
print(len(params)) #72
---------------------------------------------------------------
from  bs4  import  BeautifulSoup
import  urllib.request   # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈
import time
```

```
params = []
for i in range(1,4):
    time.sleep(5) #여러명이 동시에 접속하면 이렇게 되기 때문에 sleep을 걸어주는 것이 예의다.
    list_url = 'https://futurechosun.com/page/'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result,'html.parser')
    result1=soup.find_all('div',class_ ='elementor-post__title')
    for i in result1:
        params.append(i.find_all('a')[0].get("href"))
print(len(params))
```

#12. 조선일보 더 나은 미래 사이트는 사진을 클릭했을때도 상세기사로 가고 기사제목을 클릭했을때도 똑같이 상세기사로 가는 구조로 되어있어서 url이 상세기사 url이 중복되어서 스크롤링이 되었습니다. 그래서 params의 요소들의 중복을 제거하시오!

```
from  bs4  import  BeautifulSoup
import  urllib.request   # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

params = []
for i in range(1,4):
    list_url = 'https://futurechosun.com/page/'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result,'html.parser')
    result1=soup.find_all('div',class_ ='elementor-post__title')
    for i in result1:
        params.append(i.find_all('a')[0].get("href"))

my_result1 = set(params)     #set함수를 이용하면 중복이 제거가 된 요소들만 남는다.
my_result2 = list(my_result1) # 중복이 제거된 요소들을 다시 리스트로 감싸서 리스트화 시킨다.

print(len(my_result2)) #36
```

#13. 위의 코드들을 가지고 bs_scroll() 이라는 함수로 생성하시오!

print(len(my_result2)) -----> return my_result2로 변경하세요~~

```
from  bs4  import  BeautifulSoup
import  urllib.request   # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

def bs_scroll():

    params = []
    for i in range(1,4):
        list_url = 'https://futurechosun.com/page/'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
```

```
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result,'html.parser')
        result1=soup.find_all('div',class_ ='elementor-post__title')
        for i in result1:
            params.append(i.find_all('a')[0].get("href"))

    my_result1 = set(params)
    my_result2 = list(my_result1)

    return my_result2

print(bs_scroll())
----------------------------------------------------------------------

def bs_scroll():
    params1 = []

    for i in range(1,4):

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_ ="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)

print(bs_scroll())
```

이렇게 하면 중복제거 안해도 제대로 나온다.

#14. 더나은 미래의 상세기사 url하나를 가지고 본문 기사를 출력하시오.

기사: https://futurechosun.com/archives/52491

```
from  bs4  import  BeautifulSoup
import  urllib.request

list_url = 'https://futurechosun.com/archives/52491'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup=BeautifulSoup(result,"html.parser")
result=soup.find_all('div', class_ ='elementor-element elementor-element-24e82692
elementor-widget__width-initial elementor-widget elementor-widget-theme-post-
content')
for i in result:
```

```
    print(i.find_all('p'))
```

#16. 방금 생성한 bs_detail_scroll() 함수의 코드안에 bs_scroll()함수를 호출하는 코드를 추가해서 bs_scroll() 함수가 리턴하는 36개의 상세기사 url에 대한 기사 본문이 전부 출력되게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,4):

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_ ="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)


def bs_detail_scroll():
    list_url = bs_scroll()
    for  i  in  list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result=soup.find_all('div', class_ ='elementor-element elementor-element-24e82692
elementor-widget__width-initial elementor-widget elementor-widget-theme-post-
content')
        for i in result:
            print(i.find_all('p'))

print( bs_detail_scroll())
```

#17. bs_detail_scroll() 함수의 params2 리스트를 추가해서 params2 리스트에 지금 위에서 출력하고 있는 36개의 기사가 append되게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,3):
```

```
        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_ ="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)


def bs_detail_scroll():
    list_url = bs_scroll()
    params2 =[]
    for  i  in  list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result=soup.find_all('div', class_ ='elementor-element elementor-element-24e82692
elementor-widget__width-initial elementor-widget elementor-widget-theme-post-
content')
        for i in result:
            params2.append(i.find_all('p'))
    return params2

print( bs_detail_scroll())
```

#18. 위에서 수집한 기사가 들어있는 params2 리스트의 내용을 c드라이브 밑에 data 밑에 bongsa.txt로 생성되게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,2):

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_ ="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))
```

```
    return(params1)


def bs_detail_scroll():
    list_url = bs_scroll()
    f = open('c:₩₩data₩₩bongsa.txt', 'w', encoding="UTF-8")
    for  i  in  list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result=soup.find_all('div', class_ ='elementor-element elementor-element-24e82692
elementor-widget__width-initial elementor-widget elementor-widget-theme-post-
content')
        for i in result:
            for j in i.find_all('p'):
                f.write(str(j.get_text()) + "₩n")


    f.close()

print( bs_detail_scroll())
```

■ 이데일리 사이트 웹스크롤링 하기

#예제1. 이데일리에서 "첫눈"으로 검색한 기사들을 검색하시오!

https://www.edaily.co.kr/

https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1

#예제2. 위의 url의 첫 페이지의 html 코드를 BeautifulSoup으로 파싱하시오~
(파싱은 사람이 알아볼 수 있는 언어를 기계어로 변경하는게 파싱입니다.)
BeautifulSoup의 함수를 사용해서 웹 스크롤링할 수 있도록 변환하는 작업

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88
&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
print(soup)
```

#예제3. 상세기사 url을 찾기 위한 테그 이름과 클래스 이름을 알아내시오!

답: 테그이름은 div이고 클래스 이름은 newsbox_04 입니다.

#예제4. 위의 div 테그와 newsbox_04 클래스를 가지고 href의 상세기사 url을 출력하시오!

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
result1 = soup.find_all('div', class_ = 'newsbox_04')
for i in result1:
    print(i.find_all('a')[0])
```

이렇게 하면...

```
<a href="/news/read?newsId=01098806625997864&amp;mediaCodeNo=258" title="베일 벗은 '허쉬'... 첫방부터 안방 '들었다 놨다'">
<span class="newsbox_visual">
<img alt="베일 벗은 '허쉬'... 첫방부터 안방 '들었다 놨다'"
onerror="this.onerror=null;this.src='http://spnimage.edaily.co.kr/images/photo/files/NP/S/2020/12/PS20121200014.jpg';"
src="https://spnimage.edaily.co.kr/images/photo/files/NP/S/2020/12/PS20121200014kb.jpg"/>
</span>
<ul class="newsbox_texts">
<li>베일 벗은 '허쉬'... 첫방부터 안방 '들었다 놨다'</li>
<li>(사진=JTBC '허쉬' 방송화면)[이데일리 스타in 윤기백 기자] '허쉬'가 첫 방송부터 남다른 존재감으로 안방극장을 들었다 놨다 했다.JTBC 새 금토드라마 '허쉬'가 지난 11일, 뜨거운 호평 속에 첫 방송됐다. 신문사 '매일한국'을 배경으로 우리와 별반 다르지 않은 월급쟁이 기자들의 밥벌이 라이프를 유쾌하고 리얼하게 그려내며 시청자들을 단숨에 사로잡았다. 기대를 확신으로 바꾼 '올타임 레전드' 황정민의 귀환은 완벽했고, '믿고 보는 배우' 임윤아는 기대 이상의 변신으로 호평을 이끌었다. 이에 시청자들의 반응도 뜨거웠다. 1회 시청률은 전국 3.4%, 수도권 4.1%(닐슨코리아, 유료가구 기준)를 기록하며 기분 좋은 출발을 알렸다.'허쉬'는 누구나 공감할 수 있는 이야기를 유쾌하게 풀어냈다. 정규직 전환의 부푼 꿈을 안고 매일한국에 입성한 인턴부터, 두드려도 깨지지 않는 현실과 타협하며 오늘도 '술푼' 하루를 보내는 월급쟁이 기자들의 모습은 격한 공감을 불러왔다. 이날 1회 방송은 '밥'이라는 부제로 문을 열었다. 여느 날과 다를 것 없어 보이던 매일한국에 새로운 바람이 불고 있었다. 정기인사 결과는 단연 특종거리였다. 편집국장 나성원(손병호 분)의 비위를 맞춰가며 승진을 노리던 '아첨의 달인' 디지털 뉴스부(이하 디뉴부) 엄성한(박호산 분) 부장의 계획은 물거품이 됐고, 정세준(김원해 분) 차장은 오랫동안 몸담았던 정치부를 떠나 매일한국의 공식 유배지이자 '폭탄(?)' 처리반 디뉴부로 좌천됐다. 나국장이 건네는 위로의 건배사도 소용없었다. "저널리스트가 아닌 제너럴리스트가 돼라"는 그의 영혼 없는 뻔한 연설에, "나는 너무 너절한 너절리스트"라며 회식 자리를 발칵 뒤집어 놓은 정세준의 뼈 때리는 술주정은 웃프기까지 했다.한편 이지수와 인턴들은 65년 전통의 매일한국 입성에 마냥 들떠 있었다. 펜대보다 큐대 잡는 것이 일상이 되어 버린 '고인물' 기자 한준혁에게도 변화가 찾아왔다. 바로 인턴들의 교육을 담당하게 된 것. 한준혁과 이지수 사이에는 첫 면담부터 미묘한 불꽃이 튀었다. 한준혁은 "밥은 펜보다 강하다"는 소신 발언으로 면접장을 발칵 뒤집었다는 이지수에게 "그런 말을 하고도 졸업 첫해에 인턴 합격했으면, 금수저? 황금빽?"이라는 농담으로 그의 심기를 건드렸다. 인턴 경력도 빼곡하고 능력도 좋지만, 출신 대학 한 줄이 못내 마음에 걸리는 오수연의 이력서에는 씁쓸함을 감추지 못했다. 하지만 누구보다 잘 아는 건 바로 오수연 자신이었다. 답답한 현실과 막막한 앞날에 눈물 흘리는 그에게 한준혁은 "꺾이지 마라" 다독이면서도 그리 편치만은 않았다.바로 6년 전 '그날'의 일 때문이었다. 당시 담당 부장이었던 나성원이 조작한 가짜 뉴스로 절친했던 이용민(박윤희 분) PD가 극단적 선택을 하며, 한준혁의 기
```

자 인생을 뒤바꿔놓은 것. 특히, 억울하게 세상을 등진 이용민 PD가 이지수의 아버지였음이 밝혀져 충격을 안겼다. 이지수 손에 들린 휴대폰 속, 기사 바이라인에 적힌 '한준혁'이라는 이름 세 글자는 두 사람의 악연을 예고하며 궁금증을 높였다.'허쉬'는 시작부터 공감의 차원이 달랐다. 유쾌하게 웃다 보면 어느새 가슴 뭉클해지는 여운을 남겼다. "사람들은 우리를 기자라고 부르지만, 여기는 그냥 회사다"라는 한준혁의 내레이션처럼, 먹고 사는 문제에서 자유로울 수 없는 직장인 기자들의 고뇌는 이들이 들려줄 이야기를 더욱 궁금하게 만들었다. '기레기'를 자처하는 한준혁을 비롯해 생존과 양심, 그 경계에서 끊임없이 흔들리는 기자들의 씁쓸한 자조는 지극히 현실적이기에 더욱 깊숙이 와닿았다.황정민, 임윤아의 열연은 공감의 깊이를 더했다. 사람 냄새 진한 '한준혁' 캐릭터를 노련하게 풀어낸 황정민의 힘은 대단했다. 열정이라곤 찾아볼 수 없는 한준혁이지만, 그의 내면 어딘가에 남아있는 불씨를 불쑥불쑥 내비치는 복잡한 심경을 포착한 황정민의 연기는 이야기에 빠져들게 만들었다. 특히, 믿기 힘든 진실 앞에 감정을 폭발시키는 황정민의 열연은 가히 압권이었다. 임윤아의 연기 변신도 완벽했다. 할 말은 하고야 마는 소신과 패기의 '사이다' 매력을 발산, 또 하나의 인생 캐릭터를 탄생시켰다. 아버지의 장례식장에서 눈물을 삼키며 맨밥을 밀어 넣는 그의 열연은 시청자들의 가슴까지 저릿하게 했다. 여기에 더해진 "눈물은 아래로 떨어져도 숟가락은 위로 올라가야 하니까"라는 한준혁의 내레이션은 두 사람의 과거 사연과 함께, 앞으로의 이야기를 더욱 기대케 했다.

이런식으로 결과가 나온다.

이런식으로 나오면 .get()을 할 수 있다.

#앞의 코드들

```
result1 = soup.find_all('div', class_ = 'newsbox_04')
for i in result1:
    print("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))
```

이렇게 해주면

http://edaily.co.kr//news/read?newsId=01423526625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01439926625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01230006625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01157846625998192&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01758086625997208&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01174246625998848&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01207046625998848&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=02843766625998848&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=04057366625997208&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=03076646625999504&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01243126625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01302166625999176&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01187366625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01079126625999176&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=02499366625998520&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01289046625997536&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01453046625997536&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01216886625997864&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01098806625997864&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01387446625997536&mediaCodeNo=258

url만 뽑아낼 수 있다.

#예제5. 위의 상세기사 url을 params라는 비어있는 리스트에 담으시오!

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
result1 = soup.find_all('div', class_ = 'newsbox_04')
params = []
for i in result1:
    params.append("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))

print(params)
```

#예제6. 위의 코드들을 가지고 eda_scroll() 함수를 생성하시오!

```
from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
    list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")
    result1 = soup.find_all('div', class_ = 'newsbox_04')
    params = []
    for i in result1:
        params.append("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))

    return params

print(eda_scroll())
```

#예제7. 위의 상세기사 url하나를 가지고 기사 본문을 출력하시오!

https://www.edaily.co.kr/news/read?newsId=01387446625997536&mediaCodeNo=258

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/news/read?newsId=01387446625997536&mediaCodeNo=258'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result,"html.parser")
result1 = soup.find_all('div', class_ = 'news_body')
```

```
for i in result1:
    print(i.get_text())
```

#예제8. 위의 스크립트로 eda_detail_scroll() 함수를 생성하시오!

```
print( eda_detail_scroll() )

from bs4 import BeautifulSoup
import urllib.request

def eda_detail_scroll():
    list_url = 'https://www.edaily.co.kr/news/read?newsId=01387446625997536
&mediaCodeNo=258'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup(result,"html.parser")
    result1 = soup.find_all('div', class_ = 'news_body')
    for i in result1:
        return i.get_text()

print(eda_detail_scroll())
```

#예제9. 처음에 만들었던 함수인 eda_scroll() 함수를 수정하는데 페이지 1개가 아니라 페이지 3개의 상세기사 url인 params 리스트에 담겨지게 수정하시오.

```
from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
    params = []
    for i in range(1,4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%
88&page=1' + str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")
        result1 = soup.find_all('div', class_ = 'newsbox_04')
        for i in result1:
            params.append("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))

    return params

print(len(eda_scroll())) #60
```

#예제10. eda_detail_scroll() 함수 안에서 eda_scroll() 함수를 호출하여 상세기사 url을 60개를 다 가져와서 60개의 기사를 출력할 수 있도록 eda_detail_scroll() 함수 코드를 수정하시오.

```
from bs4 import BeautifulSoup
```

```
import urllib.request

def eda_scroll():
    params = []
    for i in range(1,4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1' + str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")
        result1 = soup.find_all('div', class_ = 'newsbox_04')
        for i in result1:
            params.append("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))

    return params

def eda_detail_scroll():
    list_url = eda_scroll()
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup(result,"html.parser")
        result1 = soup.find_all('div', class_ = 'news_body')
        for i in result1:
            print( i.get_text())

eda_detail_scroll()
```

>>> return은 한번밖에 안되기 때문에 eda_detail_scroll()에서는 return이 아닌 print()로 출력해줘야 한다.
그리고 맨 마지막에는 print() 안쓰고 그냥 eda_detail_scroll()로 해줘도 결과가 나옴

#예제11. 위에서 출력되고 있는 기사본문이 c 드라이브 밑에 data 밑에 eda.txt로 저장되게 하시오!

```
from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
    params = []
    for i in range(1,4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1' + str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")
        result1 = soup.find_all('div', class_ = 'newsbox_04')
        for i in result1:
            params.append("http://edaily.co.kr/" + i.find_all('a')[0].get("href"))
```

```
    return params

def eda_detail_scroll():
    list_url = eda_scroll()
    f = open('c:₩₩data₩₩eda.txt', 'w', encoding="UTF-8")
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup(result,"html.parser")
        result1 = soup.find_all('div', class_ = 'news_body')
        for i in result1:
            f.write(str(i.get_text()) + "₩n")

    f.close()

eda_detail_scroll()
```

■ 연합뉴스에서 인공지능으로 검색한 기사들을 스크롤링 하기

연합뉴스는 셀레니움을 이용해서 웹스크롤링을 해야합니다.
(셀레니움은 손으로 클릭해서 웹의 기사를 긁어오는 것을 컴퓨터에게 시키는 모듈이름)

 # 1. 연합 뉴스에서 인공지능으로 검색했을 때 url을 알아내시오

https://www.yna.co.kr/search/index?query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&ctype=A&page_no=1

 # 2. 위의 url의 html 코드를 beautiful soup으로 파싱하여 출력하시오!

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.yna.co.kr/search/index?query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&ctype=A&page_no=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
print(soup)
```

#3. 위의 사이트에서 상세기사 url을 알아내기 위한 테그이름과 클래스 이름을 알아내시오.

답: 테그이름은 div이고 클래스 이름은 cts_atclst 입니다.

#4. 위의 테그 이름과 클래스 이름을 가지고 상세 url을 출력하시오!

>>> 출력불가

12/18 (144-147)

2020년 12월 18일 금요일 오전 9:44

■ 144. 파이썬에서 워드 클라우드 그리기

웹스크롤링한 데이터를 분석을 하는데 분석하는 방법

1. 우리회사의 신제품이 출시되었을때 소비자들의 반응을 살펴보고자 할 때 (기업) ---> 감정분석
2. 시기별 사회현상을 파악하고자 할 때 (국가)
3. 인공지능 상담원(딥러닝의 RNN 활용)을 만들기 위한 자연어 학습 데이터로 웹스크롤링한 데이터를 활용
4. 안공지능의 눈이라고 할 수 있는 딥러닝의 CNN의 신경망의 학습 자료로 이미지 데이터가 활용

⊙ 빅데이터 기사 문제: 데이터의 구조적 관점에서 3가지로 나뉜다:

1. 정형데이터 : 정형화 된 스키마 구조. DBMS에 저장될 수 있는 구조
   예: Oracle의 emp 테이블, MySQL, MSSQL의 테이블
2. 반정형데이터 : 데이터 내부의 데이터 구조에 대한 메타정보가 포함된 구조의 데이터
   예: html 문서, xml 문서
3. 비정형데이터 : 웹스크롤링 기술로 수집해서 모은 데이터
   예: 텍스트 문서, 이미지, 동영상

■ 예제. 파이썬에서 워드 클라우드 그리기

1. 시작버튼 - > 아나콘다 프롬프트(anaconda prompt) 창을 열고 wordcloud 패키지 설치

conda install wordcloud

또는

pip install wordcloud 치고 enter 누름

2. c 드라이브 밑에 project 폴더를 생성

3. project 폴더 밑에 4가지 파일을 둡니다.

- usa_im.png
- s_korea.png
- word.txt
- 중앙일보 스크롤링했던 기사 파일 my_text21.txt

```
# 텍스트마이닝 데이터 정제

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt  # 그래프 그리는 모듈
from os import path     #  os 에 있는 파일을 파이썬에서 인식하기 위해서
import re   #  데이터 정제를 위해서 필요한 모듈
import numpy as np
```

```
from PIL import Image  # 이미지 시각화를 위한 모듈

# 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:₩₩project₩₩usa_im.png"))

# 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
script = 'sample6_laland_review.txt'

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
d = path.dirname("c:₩₩project₩₩")

# 기사 스크립트와 os 의 위치를 연결하여 utf8로 인코딩해서 한글 텍스트를
# text 변수로 리턴한다.
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()
print(text)

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
file = open('c:/project/word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') #어절별로 분리해서 word에 담아 리스트로 구성한다.
for i in word:
    text = re.sub(i,'',text)  #re.sub('있다', '', '있다') <--라라랜드 게시글에 있다를 ''으로 대체하겠다.
#설명: 일반적인 문장에서 자주나오는 단어들을 전부 ''로 대체하겠다.
#re.sub('있다','','있다')
#re.sub('하지만','','하지만')

# 워드 클라우드를 그린다.
wordcloud = WordCloud(font_path='C://Windows//Fonts//gulim', # 글씨체
                      stopwords=STOPWORDS,   # 마침표, 느낌표,싱글 쿼테이션 등을 정제
                      max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
                      background_color='white', # 배경색깔
                      max_font_size = 100, # 최대 글씨 크기
                      min_font_size = 1, # 최소 글씨
                      mask = usa_mask, # 배경 모양
                      colormap='jet').generate(text).to_file('c:/project/lala_cloud.png')
                  # c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')  # 글씨가 퍼지는 스타일
plt.axis("off") #축 표시 없음
```

⊙ 데이터 분석가가 하는일? 웹스크롤링 ---------> 시각화 (워드 클라우드)

감정분석

비정형 데이터를 활용: 기업 : 기업의 의사결정의 도움을 주기 위해서

정부 : 현 사회현상 파악

팀장님이 시킨일?

지금 우리회사 게시판에 올라온 신제품(화장품)에 대한 반응을 긍정적인 반응과 부정적인 반응을 조사해서 보고해라~

■ 145. 네이버 영화 평점을 긍정과 부정으로 분류해서 게시글 시각화 하기

⊙ 데이터 분석가가 하는일? 웹스크롤링 ---------> 시각화 (워드 클라우드)

감정분석

비정형 데이터를 활용: 기업 : 기업의 의사결정의 도움을 주기 위해서

정부 : 현 사회현상 파악

팀장님이 시킨일?

지금 우리회사 게시판에 올라온 신제품(화장품)에 대한 반응을 긍정적인 반응과 부정적인 반응을 조사해서 보고해라~

문제 431-435

■ 146. 웹에 있는 사진을 스크롤링 하는 방법 (구글 이미지)

딥러닝 기술? 1. cnn: 인공지능의 눈
예 : 관련된 학습 빅데이터? 이미지
2. rnn : 인공지능의 입과 귀
예: 관련된 학습 빅데이터? 자연어 처리를 위한 문장들

셀레니움을 써서 마치 손으로 클릭해서 이미지를 저장하듯이 저장을 하는데 컴퓨터를 시켜서 자동화 시키는 방법으로 스크롤링을 합니다.

<구글에서 이미지 스크롤링 하기>

1. 크롬 웹브라우져가 설치 되어져 있어야 합니다.
2. c 드라이브 밑에 chromedriver 폴더를 생성하고 거기에 chromedriver.exe를 넣어야 합니다.
3. c 드라이브 밑에 gimages 폴더를 생성합니다.
4. 다운받을 이미지 키워드를 결정합니다. 향수
5. 아나콘다 프롬프트 창을 열고 selenium을 설치합니다.
   conda install selenium 또는
   pip install selenium

6. 구글에서 향수로 검색했을 때 웹스크롤링 코드

```
#########################################################
import urllib.request # 웹 url을 파이썬이 인식 할 수 있게하는 패키지
from  bs4 import BeautifulSoup # html에서 데이터 검색을 용이하게 하는 패키지
from selenium import webdriver
# selenium : 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크
# 손으로 클릭하는 것을 컴퓨터가 대신하면서 스크롤링하게 하는 패키지

from selenium.webdriver.common.keys import Keys
import time        # 중간중간 sleep 을 걸어야 해서 time 모듈 import

############################ url 받아오기 ############################

# 웹브라우져로 크롬을 사용할거라서 크롬 드라이버를 다운받아 아래 파일경로의 위치에 둔다
# 팬텀 js로 하면 백그라운드로 실행할 수 있음
binary = 'c:₩chromedriver/chromedriver.exe'
```

```
# 브라우져를 인스턴스화
browser = webdriver.Chrome(binary)

# 구글의 이미지 검색 url 받아옴(아무것도 안 쳤을때의 url)
browser.get("https://www.google.co.kr/imghp?hl=ko&tab=wi&ei=l1AdWbegOcra8QXvtr-4Cw&ved=
0EKouCBUoAQ")

# 구글의 이미지 검색에 해당하는 input 창의 id 가 '  ?  ' 임(검색창에 해당하는 html코드를 찾아서 elem 사용
하도록 설정) ?: id가 뭔지 알아낼 것
# input창 찾는 방법은 원노트에 있음

# elem = browser.find_elements_by_class_name('gLFyf gsfi') # Tip : f12누른후 커서를 검색창에 올려두고 id를
찾으면 best

elem = browser.find_element_by_xpath("//*[@class='gLFyf gsfi']")  # 위의 코드대로 하거나 이렇게 하거나 둘
중 하나 select


############################ 검색어 입력 ############################

# elem 이 input 창과 연결되어 스스로 햄버거를 검색
elem.send_keys("향수") # 여기에 스크롤링하고싶은 검색어를 입력

# 웹에서의 submit 은 엔터의 역할을 함
elem.submit()

# 현재 결과 더보기는 구현 되어있지 않은상태 -> 구글의 경우 400개 image가 저장됨.

############################ 반복할 횟수 ############################

# 스크롤을 내리려면 브라우져 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번 하고 End키를 눌러야함

for i in range(1, 6): # 5번 스크롤 내려가게 구현된 상태 range(1,5)
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(10)          # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르
는데 end1번누르고 10초 쉼

time.sleep(10)                      # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다
운받아질 수 있음.)
html = browser.page_source          # 크롬브라우져에서 현재 불러온 소스 가져옴
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


browser.find_element_by_xpath("//*[@class='mye4qd']").click()  # 여기가 결과 더보기 코드입니다

for i in range(1, 5): # 4번 스크롤 내려가게 구현된 상태 range(1,5)
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(10)          # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르
는데 end1번누르고 10초 쉼

time.sleep(10)                      # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다
운받아질 수 있음.)
html = browser.page_source          # 크롬브라우져에서 현재 불러온 소스 가져옴
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정

############################ 그림파일 저장 ############################

### 검색한 구글 이미지의 url을 따오는 코드 ###
```

```
def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="rg_i Q4LuWd")  # 구글 이미지 url 이 있는 img 태그의 _img 클래스에 가서 (f12로 확인가능.)
    for im in imgList:
        try :
            params.append(im["src"])                    # params 리스트에 image url 을 담음.
        except KeyError:
            params.append(im["data-src"])
    return params

# except부분
# 이미지의 상세 url 의 값이 있는 src 가 없을 경우
# data-src 로 가져오시오 ~


def fetch_detail_url():
    params = fetch_list_url()

    for idx,p in enumerate(params,1):
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/gimages/" + str(idx) + "_google.jpg")

# enumerate 는 리스트의 모든 요소를 인덱스와 쌍으로 추출
# 하는 함수 . 숫자 1은 인덱스를 1부터 시작해라 ~

fetch_detail_url()

# 끝나면 브라우져 닫기
browser.quit()
```

딥러닝으로 인공지능을 학습시키려면 한 클래스당 2000장 정도가 있으면 적당합니다.

개와 고양이를 알아보는 인공지능을 만들고 싶으면?
개 - 2000장
고양이 -2000장 필요

■ 147. 이미지 스크롤링 하기 (네이버 이미지 검색)

```
import  urllib.request  #웹url을 파이썬이 인식할 수 있게 하는 패키지
from  bs4  import  BeautifulSoup #html 코드에서 원하는 지점을 빨리 찾을 수 있게 만든 모듈
from selenium import webdriver  #손으로 클릭하는것을 컴퓨터가 하게끔 하는 모듈
from selenium.webdriver.common.keys import Keys
import time #중간중간 sleep을 주려고 임포트 함

binary = 'C:₩chromedriver/chromedriver.exe' #크롬 드라이버 위치 지정
browser = webdriver.Chrome(binary) #browser 객체 생성
browser.get("https://search.naver.com/search.naver?where=image&amp;sm=stb_nmr&amp;")
elem = browser.find_element_by_id("nx_query") #네이버는 id로 찾음, 검색창 지점을 알아내서 elem에 담는다.
#find_elements_by_class_name("") #클래스 이름으로 찾을 때 필요한 코드
```

```
# 검색어 입력
elem.send_keys("아이언맨")  # 컴퓨터가 아이언맨 글씨를 직접 적는다.
elem.submit()                    # 그리고 엔터를 친다.

# 반복할 횟수
for i in range(1,5): #end키를 누르면서 아래로 내리는데
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5)   #슬립을 5초 주면서 5번 수행한다.

time.sleep(5) # 5초 멈췄다가

html = browser.page_source  # 현 페이지의 html 코드를 불러와서
soup = BeautifulSoup(html,"lxml")  # BeautifulSoup을 이용할 수 있도록 파싱한다.


def fetch_list_url():  #이미지의 상세 url 가져오는 함수
    params = []
    imgList = soup.find_all("img", class_="_img") #img 테그의 클래스 이름 _img로 접근
    for im in imgList:
        params.append(im["src"])  #src의 값을 가져와서 params에 append 합니다.
    return params


def  fetch_detail_url(): #상세 이미지 url 가져와서 이미지 다운로드하는 함수
    params = fetch_list_url()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/naver/"+ str(a) + ".jpg" )

        a = a + 1

fetch_detail_url()

browser.quit()
```

12/21 (148-149)

2020년 12월 21일 월요일 오전 9:47

■ 148 파이썬과 Oracle 연동하기

연동

오라클 database ---------- 파이썬 (통계구현, 시각화, 머신러닝, 업무자동화)

↓

비즈니스 데이터

(정형화된 데이터)

⊙ 오라클과 파이썬을 연동하는 이유?

1. 오라클 데이터 베이스에서 실시간으로 변하는 데이터를 csv 파일로 매번 내릴려면 자주 내려야 하므로 그냥 바로 연동합니다.
2. 파이썬의 여러 장점들을 이용해서 데이터를 분석할 수 있기 때문입니다.
   (통계구현, 시각화, 머신러닝 구현, 업무자동화)
3. 폐사진을 숫자로 변환해서 오라클 database에 저장합니다.
   이미지를 숫자로 변환해서 오라클 database에 저장합니다. db에 저장하고 관리를 합니다.
   os에 파일로 남겨만 두지 않습니다.
   db 저장시 장점 : 백업과 복구를 할 수 있다. 좀 더 효율적으로 data관리를 할 수가 있다.
   os에 파일로 저장되어 있으면 바이러스에 노출될 수도 있습니다.

⊙ 빅데이터 기사 시험: 데이터 구조에 따른 종류 3가지

1. 정형화 데이터 : rdbms에 저장된 데이터
   ↓
   relational database management system
   (관계형 데이터베이스 관리 시스템)
2. 반정형화 데이터 : html, 웹로그 데이터
3. 비정형화 데이터 : 텍스트, 동영상, 이미지 데이터
   (SNS)

⊙ 오라클과 파이썬 연동

1. **도스창을 열고 리스너의 상태를 확인합니다.**
   리스너 상태를 확인하는 명령어: lsnrctl status
   ↓
   외부에서 오라클에 접속하려면 리스너를 통해야지만 접속이 됩니다.
   리스너가 접속을 허용해 주어야 접속이 되는 것입니다.

리스너가 가지고 있는 정보 중 3가지를 확인해야합니다.

a. ip주소 : (건물주소) localhost
b. 포트번호 : (건물안의 복도) : 1521
c. 서비스 이름 : (회사 이름) : xe 혹은 orcl 혹은 orcl2

2. **위의 정보들을 이용해서 오라클 database에 접속을 해봅니다.**

sqlplus scott/tiger@localhost:1521/orcl

3. **아나콘다 프롬프트창을 열고 cx_Oracle 모듈을 설치합니다.**
conda install cx_Oracle

4. **파이썬에서 오라클과 연동하는 코드를 작성한다.**
(spyder에서 작성하세요~)

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
print(dsn)
```

결과:

```
runcell(0, 'C:/Users/admin/.spyder-py3/temp.py')
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
(CONNECT_DATA=(SID=orcl)))
##############################################
```

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select*from emp""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

결과:

```
     0       1          2       3          4       5       6   7
0  7839    KING  PRESIDENT     NaN 1981-11-17  5000.0     NaN  10
1  7698   BLAKE    MANAGER  7839.0 1981-05-01  2850.0     NaN  30
2  7782   CLARK    MANAGER  7839.0 1981-05-09  2450.0     NaN  10
3  7566   JONES    MANAGER  7839.0 1981-04-01  2975.0     NaN  20
4  7654  MARTIN   SALESMAN  7698.0 1981-09-10  1250.0  1400.0  30
```

```
5   7499   ALLEN   SALESMAN  7698.0 1981-02-11  1600.0   300.0  30
6   7844  TURNER   SALESMAN  7698.0 1981-08-21  1500.0     0.0  30
7   7900   JAMES      CLERK  7698.0 1981-12-11   950.0     NaN  30
8   7521    WARD   SALESMAN  7698.0 1981-02-23  1250.0   500.0  30
9   7902    FORD    ANALYST  7566.0 1981-12-11  3000.0     NaN  20
10  7369   SMITH      CLERK  7902.0 1980-12-09   800.0     NaN  20
11  7788   SCOTT    ANALYST  7566.0 1982-12-22  3000.0     NaN  20
12  7876   ADAMS      CLERK  7788.0 1983-01-15  1100.0     NaN  20
13  7934  MILLER      CLERK  7782.0 1982-01-11  1300.0     NaN  10
```

맥은 import 전에
Location = r"/" #다운받은 위치 써주기

■ 149. 파이썬과 mySQL 연동

현업에서는 오라클도 많이 쓰지만 mySQL도 많이 사용합니다.
오라클은 가격이 많이나가서 중요한 데이터(Business data)는 오라클에 저장하고 상대적으로 덜 중요한 데이터는 mySQL에 저장합니다.

⊙ mySQL을 실행하고 database를 생성하는 방법

1. mysql> create database orcl;
2. mysql> use orcl;
   Database changed

⊙ mySQL에서 emp테이블 생성하는 방법

```
Create table emp
(empno int(4) not null,
Ename varchar(10),
Job varchar(9),
Mgr int(4),
Hiredate date,
Sal int(7),
Comm int(7),
Deptno int(4) );

create table dept
(deptno int(2),
dname varchar(20),
loc varchar(20) );

INSERT INTO dept VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept VALUES (30,'SALES','CHICAGO');
INSERT INTO dept VALUES (40,'OPERATIONS','BOSTON');

INSERT INTO emp VALUES (7839,'KING','PRESIDENT',NULL,'81-11-17',5000,NULL,10);
INSERT INTO emp VALUES (7698,'BLAKE','MANAGER',7839,'81-05-01',2850,NULL,30);
INSERT INTO emp VALUES (7782,'CLARK','MANAGER',7839,'81-05-09',2450,NULL,10);
```

```
INSERT INTO emp VALUES (7566,'JONES','MANAGER',7839,'81-04-01',2975,NULL,20);
INSERT INTO emp VALUES (7654,'MARTIN','SALESMAN',7698,'81-09-10',1250,1400,30);
INSERT INTO emp VALUES (7844,'TURNER','SALESMAN',7698,'81-08-21',1500,0,30);
INSERT INTO emp VALUES (7900,'JAMES','CLERK',7698,'81-12-11',950,NULL,30);
INSERT INTO emp VALUES (7521,'WARD','SALESMAN',7698,'81-02-23',1250,500,30);
INSERT INTO emp VALUES (7902,'FORD','ANALYST',7566,'81-12-11',3000,NULL,20);
INSERT INTO emp VALUES (7369,'SMITH','CLERK',7902,'80-12-09',800,NULL,20);
INSERT INTO emp VALUES (7788,'SCOTT','ANALYST',7566,'82-12-22',3000,NULL,20);
INSERT INTO emp VALUES (7876,'ADAMS','CLERK',7788,'83-01-15',1100,NULL,20);
INSERT INTO emp VALUES (7934,'MILLER','CLERK',7782,'82-01-11',1300,NULL,10);
commit;
```

select * from emp;
하면

결과:

```
+-------+--------+-----------+------+------------+------+------+--------+
| empno | Ename  | Job       | Mgr  | Hiredate   | Sal  | Comm | Deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-05-09 | 2450 | NULL |     10 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-01 | 2975 | NULL |     20 |
|  7654 | MARTIN | SALESMAN  | 7698 | 1981-09-10 | 1250 | 1400 |     30 |
|  7844 | TURNER | SALESMAN  | 7698 | 1981-08-21 | 1500 |    0 |     30 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-11 |  950 | NULL |     30 |
|  7521 | WARD   | SALESMAN  | 7698 | 1981-02-23 | 1250 |  500 |     30 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-11 | 3000 | NULL |     20 |
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-09 |  800 | NULL |     20 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-22 | 3000 | NULL |     20 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-15 | 1100 | NULL |     20 |
|  7934 | MILLER | CLERK     | 7782 | 1982-01-11 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
```

이렇게 뜬다~~

■ Oracle과 MySQL 함수 비교

	Oracle	vs	MySQL
1.	nvl		ifnull
2.	sysdate		sysdate()
3.	months_between		period_add
4.	decode		if
5.	rollup		with rollup
6.	listagg		group_concat

MySQL을 켰을 때 가장 먼저 확인할 것:
자동커밋이 활성화 되어있는지!!!!

select @@autocommit;
auto commit 이 1이라면 켜져있는 거라서

set autocommit = FALSE;
해서 꺼준다!!!

12/22 (150-153)

2020년 12월 22일 화요일 오전 9:50

■ 150. 이미지를 숫자로 변환하는 방법 (폐사진)

딥러닝 수업을 할 때 이미지를 인공신경망에 넣어서 학습을 시키는데 인공 신경망에 사진을 넣을 때 숫자로 변환을 해서 넣어야 합니다.

c드라이브 밑에 images라는 폴더에 폐사진 20장을 가져다 둡니다.

사진 ----------> 인공신경망 ----------> 폐결절
정상폐
폐암
:
7~10가지

공항 검색대에 입국할 때 위반되는 물건을 반입하려고 할 때 못하게 막기위해 사람이 모니터링을 하는데 이것을 딥러닝 기술으로 컴퓨터가 알아내게끔 하려는 기술을 구현

예제1. c드라이브 밑에 images 폴더에 있는 사진들의 이름을 불러오는 함수를 생성합니다.

```
import os
test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    return file_list

print(image_load(test_image))
```

결과:
['1.png', '10.png', '11.png', '12.png', '13.png', '14.png', '15.png', '16.png', '17.png', '18.png', '19.png', '2.png', '20.png', '3.png', '4.png', '5.png', '6.png', '7.png', '8.png', '9.png']

예제2. 위의 결과에서 숫자만 나오게 함수의 코드를 수정하시오!

```
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
```

```
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것('')인
null로 변경해라~
        file_name.append(a)
    return file_name
```

결과:
['1', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '2', '20', '3', '4', '5', '6', '7', '8', '9']

예제3. 폐사진 이미지를 숫자로 변환하기 위하여 필요한 파이썬 모듈을 install 하시오!
아나콘다 프롬프트 창을 엽니다.
conda install opencv

에러가 나면 아래와 같이 하세요!
pip install opencv-python

설명 : opencv 모듈은 이미지를 파이썬에서 숫자로 변환하고 다양한 이미지 처리를 할 수 있게 해주는 기술을 제공해주는 함수

예 : 구글지도나 카카오 지도, 네이버 지도에 보면 street view가 있는데 거기에 사람 얼굴이나 자동차 번호판을 모자이크 처리를 해줘야 합니다.

예제4. 위에서 설치한 opencv 모듈을 이용해서 폐사진을 숫자로 변환한다.

```
import cv2 #opencv 모듈을 임포트 하겠다.
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것('')인
null로 변경해라~
        file_name.append(int(a))
        file_name.sort()

    file_res = []
    for k in file_name:
        file_res.append('c:₩₩images₩₩' + str(k)+'.png')

    image = []
```

```
    for h in file_res:
        img = cv2.imread(h) #이미지를 숫자로 변환하는 코드
        image.append(img)

    return image

print(image_load(test_image))
```

결과:

```
  [ 57,  57,  57],
        [ 56,  56,  56]],

       [[103, 103, 103],
        [103, 103, 103],
        [103, 103, 103],
        ...,
        [ 76,  76,  76],
        [ 78,  78,  78],
        [ 79,  79,  79]],
```

해서 담김

예제5. 위의 숫자로 변환한 리스트를 신경망에 넣기 위해서는 numpy 모듈의 array형태로 제공을 해줘야 합니다. 위의 리스트를 numpy array로 변환합니다.

```
import numpy as np #행렬 연산(신경망)을 쉽고 빠르게 할 수 있게 해주는 모듈
import cv2 #opencv 모듈을 임포트 하겠다.
import os
import re #데이터 정제를 전문으로 하는 모듈


test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것('')인
null로 변경해라~
        file_name.append(int(a))
        file_name.sort()

    file_res = []
    for k in file_name:
        file_res.append('c:₩₩images₩₩' + str(k)+'.png')

    image = []
    for h in file_res:
```

```
        img = cv2.imread(h) #이미지를 숫자로 변환하는 코드
        image.append(img)

    return np.array(image, dtype = object)

print(image_load(test_image))
```

■ 151. 이미지를 숫자로 변환하는 방법 (개와 고양이)

개와 고양이 사진을 분류하는 인공신경망을 만들려면 개와 고양이 사진을 각각 2000장씩 내려 받아서 숫자로 변환하는 작업을 해줘야 합니다.

예제1. 머신러닝 데이터 분석의 세계대회인 케글에서 캐와 고양이 사진을 내려 받습니다.
구글에서 kaggle로 검색하세요~

kaggle 검색에서 cat and dog

예제2. 개사진 30장만 c드라이브 밑에 images2라는 폴더에 넣으세요.

예제3. c드라이브 밑에 images2라는 폴더에 있는 이미지 이름을 가져오는 함수를 image_load2라는 함수로 생성하시오!

```
import os

path = 'c:₩₩images2'

def image_load2(path):
    file_list = os.listdir(path)
    return file_list

print(image_load2(path))
```

결과:
['dog.1.jpg', 'dog.10.jpg', '.11.jpg', 'dog.12.jpg', 'dog.13.jpg', 'dog.14.jpg', 'dog.15.jpg', 'dog.16.jpg', 'dog.17.jpg', 'dog.18.jpg', 'dog.19.jpg', 'dog.2.jpg', 'dog.20.jpg', 'dog.21.jpg', 'dog.22.jpg', 'dog.23.jpg', 'dog.24.jpg', 'dog.25.jpg', 'dog.26.jpg', 'dog.27.jpg', 'dog.28.jpg', 'dog.29.jpg', 'dog.3.jpg', 'dog.30.jpg', 'dog.4.jpg', 'dog.5.jpg', 'dog.6.jpg', 'dog.7.jpg', 'dog.8.jpg', 'dog.9.jpg']

예제4. 위의 개사진 이름에서 숫자만 출력하시오!

```
import os

path = 'c:₩₩images2'

def image_load2(path):
    file_list = os.listdir(path)
    file_name = []
```

```
    for i in file_list:
        a = re.sub('[^0-9]', '', i)
        file_name.append(int(a))
    return file_name

print(image_load2(path))
```

결과:
[1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 3, 30, 4, 5, 6, 7, 8, 9]

예제5. file_list의 리스트 요소가 ascending하게 정렬되게 하시오

```
import os

path = 'c:₩₩images2'

def image_load2(path):
    file_list = os.listdir(path)
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]', '', i)
        file_name.append(int(a))
        file_name.sort()
    return file_name

print(image_load2(path))
```

예제6. 아래와 같이 절대경로와 확장자가 붙어서 출력되게 하시오.

결과:
['c:₩₩images2₩₩dog1.jpg', 'c:₩₩images2₩₩dog2.jpg', 'c:₩₩images2₩₩dog3.jpg', 'c:₩₩images2₩₩dog4.jpg', 'c:₩₩images2₩₩dog5.jpg', 'c:₩₩images2₩₩dog6.jpg', 'c:₩₩images2₩₩dog7.jpg', 'c:₩₩images2₩₩dog8.jpg', 'c:₩₩images2₩₩dog9.jpg', 'c:₩₩images2₩₩dog10.jpg', 'c:₩₩images2₩₩dog11.jpg', 'c:₩₩images2₩₩dog12.jpg', 'c:₩₩images2₩₩dog13.jpg', 'c:₩₩images2₩₩dog14.jpg', 'c:₩₩images2₩₩dog15.jpg', 'c:₩₩images2₩₩dog16.jpg', 'c:₩₩images2₩₩dog17.jpg', 'c:₩₩images2₩₩dog18.jpg', 'c:₩₩images2₩₩dog19.jpg', 'c:₩₩images2₩₩dog20.jpg', 'c:₩₩images2₩₩dog21.jpg', 'c:₩₩images2₩₩dog22.jpg', 'c:₩₩images2₩₩dog23.jpg', 'c:₩₩images2₩₩dog24.jpg', 'c:₩₩images2₩₩dog25.jpg', 'c:₩₩images2₩₩dog26.jpg', 'c:₩₩images2₩₩dog27.jpg', 'c:₩₩images2₩₩dog28.jpg', 'c:₩₩images2₩₩dog29.jpg', 'c:₩₩images2₩₩dog30.jpg']

답:

```
import os

path = 'c:₩₩images2'

def image_load2(path):
    file_list = os.listdir(path)
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]', '', i)
```

```
        file_name.append(int(a))
        file_name.sort()

    file_res = []
    for k in file_name:
        file_res.append('c:₩₩images2₩₩dog' + str(k) + '.jpg')

    return file_res

print(image_load2(path))
```

예제7. 위의 개사진 이미지들을 openv와 numpy를 이용해서 숫자로 변환하고 numpy array로 반환되게 하시오!

```
import os
import cv2
import numpy as np

path = 'c:₩₩images2'

def image_load2(path):
    file_list = os.listdir(path)
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]', '', i)
        file_name.append(int(a))
        file_name.sort()

    file_res = []
    for k in file_name:
        file_res.append('c:₩₩images2₩₩dog.' + str(k) + '.jpg')

    image = []
    for h in file_res:
        img = cv2.imread(h)
        image.append(img)

    return np.array(image)
print(image_load2(path))
```

■ 152. 필수 알고리즘1 (합성곱 연산)

⊙ 알고리즘 문제를 풀어야 하는 이유?

1. 지금까지 배운 파이썬 문법을 완성하는 단계 - 알고리즘 문제 해결
2. 구글, 네이버, 카카오와 같은 대기업에 지원하고자 한다면 알고리즘 문제를 많이 풀어봐야 합니다.

   ↓

   프로그래머스 사이트에 여러 알고리즘 문제들이 올라와 있음

하루에 하나씩 꾸준히 풀면서 실력을 천천히 쌓아 올리면 됩니다.

⊙ 합성곱 연산 알고리즘?
딥러닝의 필수 알고리즘(퍼셉트론, 합성곱 연산 알고리즘)
이미지의 형상을 무시하지 않고 이미지를 그대로 인공 신경망이 학습할 수 있게 해준 수학 행렬 연산입니다.
합성곱에서는 원본 이미지는 학습해야할 데이터(사진)이고 필터(filter)는 원본 이미지에서 특징을 잡아내는데 사용되는 행렬입니다.
특징을 잡아서 feature map을 생성하여 원본이미지의 형태를 이해하는 것을 합성곱 연산이라고 합니다.

설명 : numpy란?
python 언어에서 기본적으로 지원하지 않는 배열(array) 혹은 행렬(matrix)의 계산을 쉽게 해주는 라이브러리 입니다. 머신러닝에서 많이 사용하는 선형대수학에 관련된 수식들을 python에서 쉽게 프로그래밍 할 수 있게 해줍니다.

■ 153. 필수 알고리즘2 (이진 탐색)

⊙ 이진탐색?
정렬된 데이터를 좌우 둘로 나눠서 원하는 값의 탐색범위를 좁혀가며 찾는 방법
(코딩할 때 필수!!!!)

12/23 (154-157)

2020년 12월 23일 수요일 오전 9:56

■ 154. 필수 알고리즘3 (버블정렬)

버블정렬이란?
서로 인접한 두 요소의 크기를 서로 비교하여 순서에 맞지 않는 요소를 인접한 요소와 서로 교환하여 정렬하는 정렬방법을 버블(bubble)정렬이라고 합니다.

■ 155. 필수 알고리즘4 (탐욕알고리즘)

머신러닝 배울 때 의사결정트리를 구현할 때 사용하는 알고리즘입니다.

탐욕 알고리즘이란?
탐욕 알고리즘은 매 순간마다 최선의 선택을 하는 것입니다. 선택할 때마다 가장 좋다고 생각되는 것을 선택해 나가며 최종적인 해답을 구하는 알고리즘입니다.
이 알고리즘을 설계할 때 주의할 점은 전체를 고려하는게 아니라 문제를 부분적으로 나누어, 나누어진 문제에 대한 최적의 해답을 구하게끔 해야한다는 점입니다.

예: 14원의 잔돈을 줘야하는데 잔돈의 종류가 10원, 7원, 1원이 있으면 잔돈을 가장 빨리 줄 수 있는 방법은?

> 답:7원짜리 2개를 주면 된다.
>
> 탐욕알고리즘은 10원 1개, 7원 0개, 1원 4개를 준다.

■ 156. 필수 알고리즘5 (재귀알고리즘)

재귀 알고리즘은 처음에는 이해하기가 어려운 알고리즘이지만 많이 연습해서 잘 알아두면 loop문을 최소화 하면서 코드를 간단하게 작성할 수 있는 알고리즘 입니다.

1. **재귀함수란? 반복문 + 스택구조가 결합된 함수입니다.**
   ↓
   후입선출(나중에 들어간게 먼저 나오는 구조)
   ↓
   동영상 감상

2. **재귀함수의 특징?**
   재귀함수는 함수 내에서 다시 자기 자신을 호출한 후 그 함수가 끝날 때까지 함수 호

출 이후의 명령문을 수행하지 않습니다.

3. 함수내에서 다른 함수를 호출하는 예제

```
def hap(a,b):
    return(a+b)

def gob(a,b):
    return(a*b)

def hap_gob(a,b):
    k = hap(a,b)
    m = gob(a,b)
    return k+m

print(hap_gob(2,3))
```

함수 내에서 함수를 호출

4. 숫자를 입력하면 1부터 해당 숫자까지 합을 출력하는 함수를 생성하시오.

```
print(add_func(5))
```

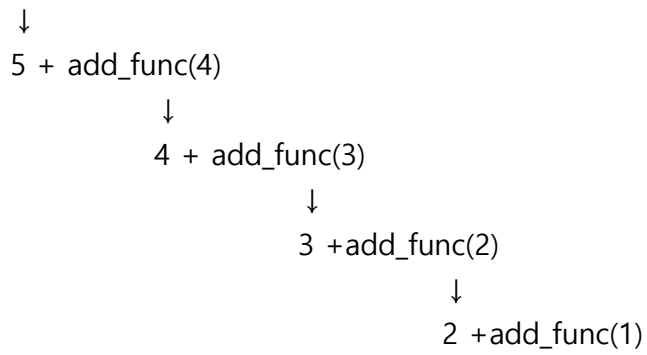
답:

```
def add_func(n):
    a = input('숫자를 입력하세요')
    cnt = 0
    for i in range(1, n+1):
        cnt +=i
    return cnt

print(add_func(5))
```

5. 위의 add_func 함수를 재귀함수로 구현하시오!

(재귀함수를 사용하면 loop문을 사용하지 않아도 됩니다.)

```
def add_func(n):
    if n == 0:
        return 0
    else:
        return n + add_func(n-1)

print(add_func(5))
```



↓
1 + add_func(0)

■ 157. 필수 알고리즘6 (LRU 알고리즘) (카카오 시험문제에 나옴)

↑
Least Recent Used

LRU 알고리즘이란 Oracle DATABASE의 메모리 관리를 효율적으로 하기 위해 고안된 대표적인 알고리즘으로 최신 데이터를 메모리에 유지시키고 오래된 데이터는 메모리에서 내보내게 하는 알고리즘

메모리에서 기록을 조회하게 되면 1초가 걸리는데 (훨씬 빠름) --->cache hit(메모리에서 데이터 조회
메모리에 없어서 디스크에서 기록을 조회하면 5초정도 걸린다. ----> cache miss (디스크에서 데이터 조회)

한번 디스크에서 읽은 데이터를 메모리에 올려 놓고 메모리에서 빠르게 데이터를 조회할 수 있도록 LRU 알고리즘을 구현해서 만든 소프트웨어 입니다.
그런데 이 메모리 공간이 한정된 공간이다보니 무한히 데이터를 올릴 수 없어서 오래된 데이터를 매모리에서 빠져나가게 되고 최신 데이터가 메모리로 그 빠져나간 자리에 올라가게 됩니다.
최근에 내가 검색한 데이터는 다시 검색할 확률이 높은 데이터이므로 메모리에 오래 두도록 하고 예전에 검색한 데이터는 메모리에서 빠져나가게 합니다. (선입선출)

⊙ 스택: 후입선출
⊙ 큐 : 선입선출

12/24 (158)

2020년 12월 24일 목요일 오전 9:55

■ 알고리즘 문제를 해결하는 순서

1. **문제를 2번 정독한다. (하고자 하는게 무엇인지 질문을 명확하게 한다)**
   a. **LRU 알고리즘에 대한 이해가 있어야 합니다.**
   b. **함수를 생성해야 하는데 입력값이 2개입니다.**
   (도시이름, 캐쉬 사이즈)
   Ex 함수의 입력값(도시이름, 4) --> 메모리 4칸
   도시이름 정하기:
   a = ['jeju', 'pangyo', 'new york', 'new york']

	jeju	pangyo	new york


<- 디스크

   결과: 5초 + 5초 + 5초 + 1초
   마지막이 1초 걸린 이유는 나머지는 다 디스크에서 데이터를 가져왔지만 마지막 new york은 이미 메모리에 있어서 1초밖에 걸리지 않음

   cache hit --> 메모리에 데이터가 있어서 메모리에서 읽은 것(1초)
   cache miss --> 메모리에 데이터가 없어서 메모리로 올림

   c. **함수의 출력값은 총 실행시간(초) 입니다.**

   질문:
   cities = ['Jeju', 'Pangyo', 'New York', 'new york']
   print(cacheProcess(cities, 4))
   결과: 16
   이 나오게 하는 방법은?

2. **해결방법을 노트에 번호 순서대로 글로 글을 적습니다.**
   a. **도시 이름은 대소문자를 구분하지 않게 만든다.**
   코딩예제: 아래의 리스트를 받아서 아래의 결과를 출력하시오!
   cities = ['Jeju', 'Pangyo', 'New York', 'new york']

   결과: ['jeju', 'pangyo', 'new york', 'new york']

   답:
   cities = ['Jeju', 'Pangyo', 'New York', 'new york']

```
city = []
for i in cities:
    city.append(i.lower())

print(city)
```

위의 comprehension을 사용해서 간단하게 수행하시오!

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']
city = [i.lower() for i in cities]
print(city)
```

⊙ **comprehension 안쓴 코드 :**

```
for i in range(1,21):
    if i%2 ==1:
        a.append(i)
print(a)
```

❖ 설명: 위의 코드는 comprehension 코드로 위의 5줄 코드를 3줄로 간소화한 코드입니다.

문법 : [출력 표현식 for 요소 in 입력시퀀스 if 조건식]

i.lower() for i in cities

예 : 1부터 20까지의 홀수를 담는 리스트를 만드시오~

```
a = [i for i in range(1,21) if i%2 ==1]
print(a)
```

b. 메모리(cache)를 만들어야 한다. (입력 숫자만큼 cache 사이즈가 구성)

코딩예제: 아래의 리스트를 생성하시오!

[None, None, None, None]

답:

```
cache = [None for i in range(1,5)]
print(cache)
```

c. 메모리에 데이터를 올린다 (cache miss니까 5초가 걸려야 한다)

코딩예제1: 아래와 같이 cache에 data를 올리세요~

결과 : [None, None, None, 'jeju']

답:

```
cache = [None for i in range(1,5)]
del cache[0]
cache.append('jeju')
print(cache)
```

코딩예제2 : 아래와 같이 cache에 data를 올리세요!

결과 : [None, None, 'jeju', 'pangyo']

답:

```
cache = [None for i in range(1,5)]
cache.append('jeju')
del cache[0]
cache.append('pangyo')
del cache[0]
print(cache)
```

코딩예제3 : 위에서 만든 코드를 가지고 아래의 함수를 생성하시오!

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']

def cacheProcess( cities, cachesize):
    city = [i.lower() for i in cities] #도시이름 다 소문자로 데이터를 만드는 작업
    cache = [None for i in range(1,5)] #None 4개 cache 구성
    for i in range(0, len(city)): # 0, 1, 2, 3으로 루프 돌기( city 요소가 4개)
        cache.append( city[i] )
        del cache[0] #city의 요소들을 하나씩 넣으면서 cache의 0번째 요소 지움
    return cache

print(cacheProcess(cities, 4))
```

코딩예제4 : 위의 함수의 결과가 cache의 결과가 아니라 수행시간이 되게하시오!

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']
print( CacheProcess( cities, 4 ) ) #16
```

답:

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']

def cacheProcess( cities, cachesize):
    city = [i.lower() for i in cities]
    cache = [None for i in range(1,5)]
    cnt = 0
    for i in range(0, len(city)):
        if city[i] in cache:
            cnt = cnt+1
            cache.append( city[i] )
            del cache[0]

        elif city[i] not in cache:
            cnt = cnt+5
            cache.append( city[i] )
            del cache[0]

    return cnt
```

```
print(cacheProcess(cities, 4))

cities= ['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo', 'Seoul',
'NewYork', 'LA']
print( cacheProcess( cities, 3 ) ) #50


cities = ['Jeju', 'Pangyo', 'Seoul', 'Jeju', 'Pangyo','Seoul', 'Jeju', 'Pangyo','Seoul']
print(cacheProcess(cities,3)) #21
```

d. 메모리에서 데이터를 찾으면 cache hit니까 1초가 걸려야 한다.

3. 번호 순서대로 글로 적은 내용을 코딩으로 번역하면 됩니다.

(한글 -----> 파이썬)

큰 문제를 작은 문제들로 쪼개서 작은 문제들을 하나씩 해결해가면서 큰 문제를 해결하는 방법으로 코딩을 해야합니다.

파이썬 문법 150개 학습 ---> 파이썬 문법을 활용하는 알고리즘 문제
(함수를 만들어라~)

⊙ LRU 알고리즘 문제를 파이썬으로 구현하는 순서
(큰 문제를 작은 문제들로 나눠서 해결하겠다.)

1. 문제를 2번 정독한다.
2. 번호 순서대로 해결방법을 기술한다. (노트나 메모장)
3. 번호 순서대로 기술한 한글로 적은 해결방법을 파이썬으로 번역한다.

■ 카카오 블라인드 코딩 시험 (LRU 알고리즘)

캐시(난이도: 하)

지도개발팀에서 근무하는 제이지는 지도에서 도시 이름을 검색하면 해당 도시와 관련된 맛집 게시물들을 데이터베이스에서 읽어 보여주는 서비스를 개발하고 있다.

이 프로그램의 테스팅 업무를 담당하고 있는 어피치는 서비스를 오픈하기 전 각 로직에 대한 성능 측정을 수행하였는데, 제이지가 작성한 부분 중 데이터베이스에서 게시물을 가져오는 부분의 실행시간이 너무 오래 걸린다는 것을 알게 되었다.

어피치는 제이지에게 해당 로직을 개선하라고 닦달하기 시작하였고, 제이지는 DB 캐시를 적용하여 성능 개선을 시도하고 있지만 캐시 크기를 얼마로 해야 효율적인지 몰라 난감한 상황이다.

어피치에게 시달리는 제이지를 도와, DB 캐시를 적용할 때 캐시 크기에 따른 실행시간 측정 프로그램을 작성하시오.

입력 형식

- 캐시 크기(cacheSize)와 도시이름 배열(cities)을 입력받는다.
- cacheSize는 정수이며, 범위는 0 ≦ cacheSize ≦ 30 이다.
- cities는 도시 이름으로 이뤄진 문자열 배열로, 최대 도시 수는 100,000개이다.
- 각 도시 이름은 공백, 숫자, 특수문자 등이 없는 영문자로 구성되며, 대소문자 구분을 하지 않는다. 도시 이름은 최대 20자로 이루어져 있다.

출력 형식

- 입력된 도시이름 배열을 순서대로 처리할 때, "총 실행시간"을 출력한다.

조건

- 캐시 교체 알고리즘은 LRU(Least Recently Used)를 사용한다.
- cache hit일 경우 실행시간은 1이다. (1초)
- cache miss일 경우 실행시간은 5이다. (5초)

■ 158. 필수 알고리즘7 (자카드 유사도 알고리즘)

자카드 유사도는 두 문장을 각각의 집합으로 만든 뒤 두 집합을 통해 유사도를 측정하는 알고리즘 입니다.

그림:

설명 : 두 문장의 교집합이 6개, 합집합이 24개이므로 자카드 유사도는 6/24 = 0.25 가 됩니다.

예제1. 다음 A집합과 B집합의 자카드 유사도를 구하시오!
A = {1, 2, 3}　　B = {2,3,4}
J(A,B) = ?
A∩ $B = \{2,3\}$
A∪ $B = \{1,2,3,4\}$

J(A,B) = $\frac{2}{4} = 0.5$

예제2. FRANCE, FRENCH가 주어졌을 때 자카드 유사도는?

A = {FR, RA, AN, NC, CE}　　B = {FR, RE, EN, NC, CH}

J(A,B) = ?
A∩ $B = \{FR, NC\}$
A∪ $B = \{FR, RA, AN, NC, CE, RE, EN, CH\}$
J(A,B) = $\frac{2}{8} = 0.25$

■ 파이썬으로 합집합과 교집합 구현하기

1. 리스트

2. 사전형
3. 튜플형
4. 집합형

```
a = {1,2,3}
b = {2,4,5}

#1. a와 b의 합집합 구하기
result = a.union(b)
print(result1) #{1, 2, 3, 4, 5}

#2. a와 b의 교집합 구하기
result2 = a.intersection(b)
print(result2) #{2}

#3. a와 b의 자카드 유사도를 구하기
result3 - len(result2)/ len(result1)
print(result3) = #0.2
```

■ 리스트로 합집합과 교집합 구하기

```
a = [1,2,3,4]
b = [2,4,5]

#1. a와 b의 합집합 구하기


result1 = set(a+b)
print(result1) #{1, 2, 3, 4, 5}

#2. a와 b의 교집합 구하기

a = [1,2,3,4]
b = [2,4,5]
result2 = []
for i in a:
        if i in b:
          result2.append(i)

print(result2) #[2, 4]
```

■ 자카드 유사도 알고리즘 문제를 파이썬으로 구현하는 방법과 순서

1. **문제를 2번 읽으면서 문제를 정확하게 파악한다.**
   (특히 질문을 명확하게 제시해줘야 합니다.)

질문: FRANCE, FRENCH의 두 단어의 자카드 유사도는?

관련 알고리즘(자카드 유사도)의 이해가 있어야 합니다.

2. **문제를 해결하기 위해서 순서별로 해결방법을 기술한다.**
   a. **두 문장을 받아서 두개의 철자로 분리하여 리스트에 저장한다.**
      문자로 된 글자 쌍만 유효하다.

```
str1 = input('문자열을 입력하세요~').upper()
str2 = input('문자열을 입력하세요~').upper()

str1 = 'FRANCE'
for i in range(len(str1)-1):
    print(str1[i:i+2])
```

결과:

```
FR
RA
AN
NC
CE
```

```
str1 = 'FRANCE'
res = []
for i in range(len(str1)-1):
    if str1[i].isalpha() and str1[i+1].isalpha(): #.isalpha() = 문자열
        res.append(str1[i:i+2])

print(res) #['FR', 'RA', 'AN', 'NC', 'CE']
```

예: FRANCE, FRENCH를 받아서 아래의 리스트를 만듭니다.

```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']
```

```
str1 = input('문자열을 입력하세요~').upper() #FRANCE
str2 = input('문자열을 입력하세요~').upper() #french

def str_split(string):
    res = []
    for i in range(len(string)-1):
        if string[i].isalpha() and string[i+1].isalpha():
            res.append(string[i:i+2])
    return res

print(str_split(str1)) #['FR', 'RA', 'AN', 'NC', 'CE']
print(str_split(str2)) #['FR', 'RE', 'EN', 'NC', 'CH']
```

   b. **a와 b 두개의 리스트의 교집합을 구합니다.**
      a = ['FR', 'RA', 'AN', 'NC', 'CE']
      b = ['FR', 'RE', 'EN', 'NC', 'CH']

답:

```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']

intersection = []
for i in a:
    if i in b:
        intersection.append(i)
print(intersection) #['FR', 'NC']
```

c. **a와 b 두개의 리스트의 합집합을 구합니다.**

```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']

union = set(a + b)
print(union) #{'AN', 'FR', 'RE', 'CH', 'RA', 'NC', 'CE', 'EN'}
```

d. **a와 b의 자카도 유사도를 구합니다.**

```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']

union = set(a + b)

intersection = []
for i in a:
    if i in b:
        intersection.append(i)

import math
len_i = len(intersection)
len_u = len(union)
print(math.trunc(len_i/len_u * 65536)) #16384
```

e. **a와 b가 모두 공집합일 경우에는 나눗셈이 정의 되지 않으니 따로 J(A,B) = 1로 정의한다. (문제 522번)**

3. **순서별로 정한 해결방법을 파이썬 코드로 구현한다. (한글 --> 파이썬)**

- set의 문제

```
A = [1,1,1,2,2,3]
B = [1,1,2,2,4,5]
result = set(A+B) #합집합
print(result) #{1, 2, 3, 4, 5}
```

우리가 원하는 합집합은 { 1,1,1,2,2,3,4,5 } 이다.
set은 중복을 다 날려버린다.

12/28 (159-160)

2020년 12월 28일 월요일 오전 9:47

■ 159. 필수 알고리즘8 (비밀지도)

⊙ 알고리즘 문제 풀 때 해결하는 순서

1. 문제를 2번 정독하면서 질문을 정확하게 이해하고 출력이 뭐가 되는지 정확하게 이해해야 합니다.

   n = 5
   arr1 = [9, 20, 28, 18, 11]
   arr2 = [30, 1, 21, 17, 28]

   secretmap(arr1,arr2,n)

   결과:

   ['######']
   ['### #']
   ['## ##']
   [' #### ']
   [' #####']
   ['### # ']

2. 풀이를 번호 순서대로 한글로 적는다. (메모)
   a. 십진수를 이진수로 변환한다.
   b. 두개의 이진수를 가지고 #과 공백을 나타낸다.

3. 한글 ----> 파이썬으로 번역하라!

■ 160. 통계 마지막 문제 (도박사 이야기)

※ 원본은 통계학(출판사 성안당) 책 78페이지

해외여행 중 어느 뒷골목에서 동전으로 내기를 하는 도박사를 만났다.
앞면이 나오면 당신에게 1달러, 뒷면이 나오면 나에게 1달러, 어때요? 이 내기 한번 해볼래요?
그런데 동전에 부정이 있는 듯해 실제로 알아봤더니 8회중 7회는 뒷면이 나왔다.
이 동전은 앞면과 뒷면이 나올 확률이 다르지 않느냐? 고 했더니 도박사는 아무렇지도 않게 "그런게 아니고 우연" 이라고 대답했다.
어떻게 반론하면 좋을까?

오늘의 문제(1-147)

2020년 11월 24일 화요일 오전 10:47

1. **아래의 스크립트를 배치모드로 실행하시오!**
   {1,2,3,4,5,6,7}

   답:
   for i in {1,2,3,4,5,6,7}
   print (i)

2. **아래와 같은 결과가 나오게 spyder에서 프로그래밍 하시오.**
   2
   3
   4

   for i in [2,3,4]:
   print (i)

3. **파이썬의 예약어가 무엇이 있는지 확인하시오!**
   import keyword
   print (keyword.kwlist)

   ['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

4. **위의 예약어를 변수로 사용하면 에러가 나는지 직접 테스트 하시오!**
   false = 1

   SyntaxError: invalid syntax 라고 나온다.

5. **변수의 자료형을 확인하시오!**
   ↓
   문자형, 숫자형, 날짜형
   b = 'scott'
   print (type(b))

   <class 'str'> ---> 문자형이라는 뜻

6. **숫자형 변수의 자료형을 확인하시오!**
   a=3
   print (type(b))

<class 'int'> ---> 숫자형/ integer

7. **아래의 리스트에서 문자 k를 출력하시오!**
   mmm = ['a', 'b', 'd', 'e', 'k', 'm', 'n', 'z']

   답:
   mmm = ['a', 'b', 'd', 'e', 'k', 'm', 'n', 'z']
   print (mmm[4])

8. **동전을 던져서 앞면이 나오는지 뒷면이 나오는지 확인하시오!**
   import random # random이라는 모듈을 이 코드에서 쓰겠다.
   ↓
   특정 목적으로 만든 파이썬 코드의 집합

   coin = ['앞면', '뒷면']
   print (random.choice(coin))
   ↓
   random모듈 안의 choice라는 함수를 이용해라

9. **점심시간 문제 위에서는 동전을 던졌는데 이번에는 주사위를 던져서 주사위의 눈이 뭐가 나오는지 확인하는 파이썬 코드를 작성하시오!**
   dice = [1,2,3,4,5,6]
   print (random.choice(dice))

10. **점심시간에 했던 주사위의 눈이 6개 있는 변수를 만들고 if문을 써서 "주사위의 눈에 숫자 5가 있습니다." 라는 메세지가 출력되게 하시오!**
    dice = [1,2,3,4,5,6]
    if 5 in dice : print ('주사위의 눈에 숫자 5가 있습니다')

11. **문제 10번의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 문제 10번의 메세지가 출력되게 하시오!**
    to_number와 같은 코드 = int( )

    a = int(input('숫자를 입력하세요~'))
    dice = [1,2,3,4,5,6]
    if a in dice :
        print ('주사위의 눈에 숫자', a,'가 있습니다')

    오라클의 ||가 파이썬에서는 , 이다.

12. **주사위의 눈을 담은 dice 변수를 만들고 숫자를 물어보게해서 해당 숫자가 주사위의 눈중에 있으면 해당 숫자가 있습니다 라는 메세지가 출력되게 하고 없으면 해당숫자가 없습니다. 라는 메세지가 출력되게 하시오.**
    a = int(input ('숫자를 입력하세요~'))

```
dice = [1,2,3,4,5,6]
if a in dice:
                print ('주사위의 숫자', a, '가 있습니다')
else:
                print ('주사위의 숫', a, '가 없습니다')
```

이렇게 해줘야 함. 이렇게 안해주면 a에 넣는 숫자 대신 'a'라는 문자가 나옴

13. 숫자 두개를 아래와 같이 각각 물어보게 하고 아래처럼 메세지가 출력되게 하시오!

첫번째 숫자를 입력하세요~ 2 / 4
두번째 숫자를 입력하세요~ 3 / 3

2는 3보다 작습니다.
4는 3보다 큽니다.

```
a = int( input ('첫번째 숫자를 입력하세요~'))
b = int( input ('두번째 숫자를 입력하세요~'))
if a<b :
    print (a,'는 ',b,'보다 작습니다.')
else:
    print (a,'는 ', b, '보다 작습니다.')
```

14. ★을 5개 출력하시오!

★★★★★

```
print('★'*5)
```

15. (알고리즘 5번 문제) 아래와 같은 결과가 출력되는 파이썬 코드를 작성하시오!

숫자를 입력하세요 ~ 5
★
★★
★★★
★★★★
★★★★★

```
a = int( input ('숫자를 입력하세요~'))
for a in range(1,a+1):
    print('★'*a)  #a가 1일때는  ★
                  #a가 2일때는  ★★
                   #a가 3일때는 ★★★
                   #a가 4일때는 ★★★★
                   #a가 5일때는 ★★★★★
```

16. 주사위를 10번을 던져서 출력되는 눈을 확인하시오.

```
dice = [1,2,3,4,5,6]
a=10
for a in range(1,a+1): #1이상 a+1미만
    print(random.choice(dice))
```

또는

```
dice = [1,2,3,4,5,6]
for i in range(10):
    print(random.choice(dice))
```

17. 동전을 10번 던지세요~

```
import random # random모듈의 코드를 지금 현재 코드창에서 사용하겠다.

coin = ['앞면', '뒷면']
for i in range(10): #또는 i in range(1,11):
        print(random.choice(coin))
```

18. 동전을 10번 던졌을 때 앞면이 나오는 횟수가 어떻게 되는가?

```
coin = ['앞면', '뒷면']
cnt = 0                                  #cnt라는 변수에 0을 할당한다.
for i in range(1, 11):                   #1부터 10까지 루프문을 10번 실행한다.
    result = random.choice(coin)         #동전 던진 결과를 result 변수에 할당한다.
    if result =='앞면':
        cnt = cnt+1          #할당연산자 오른쪽부터 실행하고 실행한 결과를 왼쪽 변
수에 할당해준다.
print (cnt)                  #for문의 실행문이 아닌 for문과 위치를 맞춰서 같은 라인
에
                             print(cnt)를 쓰면은 for문이 다 완료되면 print(cnt)가 수행
됩니다.
```

앞면 cnt = cnt+1 ---> cnt가 1입니다.
뒷면
앞면 cnt = cnt+1 ---> cnt가 2입니다.
뒷면
뒷면
앞면 cnt = cnt+1 ---> cnt가 3입니다.
뒷면
뒷면
뒷면
앞면 cnt = cnt+1 ---> cnt가 4입니다.

19. **주사위를 100번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오**

```
dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,101):
    result= random.choice(dice)
    if result == 3:
        cnt = cnt+1
print(cnt)
```

20. **주사위를 1000번 던지고 주사위의 눈이 5가 나올 확률을 구하시오!**

```
dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,1001):
    result= random.choice(dice)
    if result == 5:
        cnt = cnt+1
print(cnt/1000)
```

21. **동전을 10000번 던져서 앞면이 나올 확률을 출력하시오!**

```
coin = ['앞면','뒷면']
cnt = 0
for i in range(1,10001):
    result= random.choice(coin)
    if result == '앞면':
        cnt = cnt+1
print(cnt/10000)
```

22. **(11/24 마지막 문제) (SQL 알고리즘 13번)**
주사위 하나와 동전한개를 동시에 던져서 주사위의 눈은 5가 나오고 동전은 앞면이 나올 확률은 어떻게 되는가?

```
coin = ['앞면','뒷면']
dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,100001):
    result1= random.choice(coin)
    result2= random.choice(dice)
    if result1 == '앞면' and result2 == 5:
        cnt = cnt+1
print(cnt/100000)
```

23. **(알고리즘 4번) for loop문을 이용해서 구구단 2단을 출력하시오!**

```
2 x 1 = 2
2 x 2 = 4
  :
  :
2 x 9 = 18

a = [1,2,3,4,5,6,7,8,9}
```

```
print(2*a)

for i in range(1,10): #1부터 10미만까지
        print ('2 x ', i, '=', 2*i)
```

설명 : for loop문은 특정 문장을 반복해서 수행하고 싶을 때 사용하는 파이썬 문법
위의 문법은 1부터 9까지 9번 반복하겠다.

24. (알고리즘 8번) 구구단 전체를 파이썬으로 출력하시오!

```
2 x 1 = 2
2 x 2 = 4
   :
   :
9 x 9 = 81
```

25. 주사위를 10번 던져서 주사위의 눈을 10개 출력하시오!

```
import random
dice = [1,2,3,4,5,6]
for i in range(1, 11):
    print(random.choice(dice))
```

26. 주사위를 10번 던졌을 때 짝수가 나오는 횟수를 출력하시오.

```
import random
dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,11):
        result= random.choice(dice) #주사위의 눈이 result변수에 할당이 된다
        if result % 2 == 0: #result 변수의 주사위의 눈이 짝수이면
                cnt = cnt+1 #cnt를 1씩 증가시킨다.
        print(cnt) #print의 위치를 for문과 맞추져서 cnt에 담긴 최종 결과를 출력한다.
```

27. 위의 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 작업을 5번 반복해서 짝수가 나오는 횟수가 5개가 나오게 하시오!

```
import random
dice = [1,2,3,4,5,6]
for j in range(1,6):
    cnt = 0
    for i in range(1,11):
        result= random.choice(dice)
        if result % 2 == 0:
                cnt = cnt+1
print(cnt)
```

설명 : 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 for loop문을 5번 반복하기 위해서 그 위에 for loop문을 쓰고 5번 반복하였다.

그리고 한번에 4칸 이동시키기 위해서 코드를 선택하고 탭(tab)키를 눌러서 한번에 이동시켰다. 뒤로 이동시키는 단축키는 shift + tab입니다.

28. 동전을 100번 던져서 앞면이 나올 확률을 출력하시오

```
import random
coin = ['앞면', '뒷면']
cnt = 0
for i in range(1,101):
    result = random.choice(coin)
    if result == '앞면':
        cnt = cnt+1
print(cnt/100)
```

29. 위의 동전을 100번 던져서 앞면이 나올 확률을 구하는 직업을 50번 반복해서 출력하시오!

```
import random
coin = ['앞면', '뒷면']
for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt+1
print(cnt/100)
```

30. 파이썬 리스트의 append 함수를 이용해서 위의 확률 50개를 리스트에 담으시오~

```
import random
coin = ['앞면', '뒷면']
a = [] #비어있는 리스트 a를 생성한다.
for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt+1
    a.append(cnt/100) # a 리스트에 확률을 입력한다.

print(a)  # a리스트에 담겨있는 값들을 출력한다.
```

31. 두개의 동전을 300번 던져서 동시에 둘다 앞면이 나오는 횟수를 출력하시오!

```
import random

coin1 = ['앞면', '뒷면']
coin2 = ['앞면', '뒷면']
cnt=0
```

```
for i in range(1,301):
    result1 = random.choice(coin1)
    result2 = random.choice(coin2)
    if result1 =='앞면' and result2 =='앞면':
        cnt = cnt+1
print(cnt)
```

32. x라는 리스트를 만들고 x라는 리스트에 위의 문제에서 앞면이 나오는 횟수를 x리스트에 담는데 위의 둘다 앞면이 나오는 횟수를 출력하는 for loop문을 1000번 반복해서 x리스트에 1000개 입력하시오!

```
import random

coin1 = ['앞면', '뒷면']
coin2 = ['앞면', '뒷면']

x = []
for j in range(1,1001):
    cnt=0
    for i in range(1,301):
        result1 = random.choice(coin1)
        result2 = random.choice(coin2)
        if result1 =='앞면' and result2 =='앞면':
            cnt = cnt+1
    x.append(cnt)

print(x)
```

33. 확률변수 x의 값들의 평균과 분산과 표준편차를 출력하시오!

```
import random
import numpy as np #넘파이 모듈을 이 코드에서 사용
coin1 = ['앞면', '뒷면']
coin2 = ['앞면', '뒷면']

x = []
for j in range(1,1001):
    cnt=0
    for i in range(1,301):
        result1 = random.choice(coin1)
        result2 = random.choice(coin2)
        if result1 =='앞면' and result2 =='앞면':
            cnt = cnt+1
    x.append(cnt)

print(np.mean(x)) #75.151
print(np.var(x)) #59.364100
print(np.std(x)) #7.704816610406765
```

34. 한개의 주사위를 360번 던져서 3의 배수의 눈이 나오는 횟수를 구하는것을 1000번 하

고 그 1000개를 X라고 했을 때 이 X의 평균, 분산, 표준편차를 출력하시오.

(점심시간 문제)

```
import numpy as np

dice = [1,2,3,4,5,6]  #x라는 비어있는 리스트 생성

x = [ ]
cnt = 0
for j in range(1,1001): #100번 반복수행하는 for loop문 작성
   cnt = 0  #cnt에 0을 할당합니다.
   for i in range(1,361):  # 360번 반복수행하는 for loop문 작성
      result = random.choice(dice) # 주사위에서 눈을 하나 출력해서 result에 담는다.
      if result % 3 == 0: #그 주사위의 눈을 3으로 나눈 나머지 값이 0이면
         cnt = cnt+1 #cnt를 1씩 증가시키겠다.
   x.append(cnt)    #주사위를 360번 던져서 3의 배수가 나온 횟수가 x리스트에 추가
                    #됩니다.

print(np.mean(x))



print(np.var(x))
print(np.std(x))
```

35. **숫자 1번부터 10번까지 출력하시오!**

```
for i in range(1,11):
      print(i)
```

36. **위의 결과에서 숫자 5만 출력하지 않고 나머지 숫자만 출력하시오!**

```
for i in range (1,11):
      if i ==5:
            continue:
      print(i)
```

37. **(알고리즘 문제2) 1부터 10까지의 숫자를 출력하되 짝수만 출력하시오!**

```
for i in range(1,11):
    if i%2 == 1:    #i가 홀수이면
       continue    #재껴라~
    print(i)
```

38. **(알고리즘 문제1) 1부터 10까지의 합을 파이썬으로 구현하시오!**

```
cnt = 0
for i in range(1,11):
    cnt = cnt + i
print(cnt)
```

또는

```
x = []
for i in range(1,11):
    x.append(i)
print(sum(x))
```

39. 1부터 100번까지 출력하는 for loop문을 작성하는데 다음과 같이 숫자를 물어보게해서 입력된 숫자까지만 출력되게하시오!

숫자 입력하세요~

```
a = int( input ('숫자를 입력하세요~'))
for i in range(1,101):
    print (i)
    if i == a:
        break
```

40. (알고리즘 문제 19번) 두 숫자를 각각 물어보게하고 입력받아 두 숫자의 최대공약수를 출력하시오! (오늘의 마지막 문제)

힌트 : 16%i ==0 and 24%i==0

답:

```
num1 = int(input('첫번째 숫자를 입력하세요'))
num2 = int(input('두번째 숫자를 입력하세요'))

for i in range(min(num1,num2),0,-1):     #더 적은 숫자부터 나눗셈을 시작하도록 min으로 수정함
    if num1%i == 0 and num2%i == 0:
        break
print('최대공약수는 ',i,'입니다.')
```

41. 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자만큼 1번부터 숫자가 출력되게 하시오.

```
a = int( input ('숫자를 입력하세요~'))

for i in range(1, a+1):
        print(i)
```

42. 41번에 입력한 숫자가 다 출력되면 아래에 perfect이란 단어가 출력되게하시오!

```
a = int( input ('숫자를 입력하세요~'))

for i in range(1, a+1):
        print(i)
else:
    print('perfect')
```

43. 위의 코드를 수정해서 중단할 숫자를 또 물어보게 해서 아래와 같이 실행되게 하시오!

```
a = int( input ('숫자를 입력하세요~'))
b= int( input ('중단할 숫자를 입력하세요~'))

for i in range(1, a+1):
    print(i)
    if i == b:
        break
else:
    print('perfect')
```

44. 아래와 같이 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오. (while loop문으로 하세요)

```
숫자를 입력하세요 ~ 5
★
★★
★★★
★★★★
★★★★★
a = int( input ('숫자를 입력하세요~'))
x=0
while x < a:        0<5
    x=x+1           x=1
    print('★'*x)     1*★
```

45. 아래와 같이 결과가 출력되게 하시오!

```
숫자를 입력하세요~ 5
★★★★★
★★★★
★★★
★★
★
a = int( input ('숫자를 입력하세요~'))
x=0
while x < a:
   x=x+1
   print('★'*(a-x+1))
```

또는

```
a = int( input ('숫자를 입력하세요~'))
while x > a:
    print('★'*(a-x+1))
    b=b-1
```

46. 주사위 2개를 동시에 던져서 두개의 주사위의 눈의 합을 구하는 실행문을 10000번 반복해서 그 10000개를 비어있는 a리스트에 넣으시오. (while 루프문으로 하세요)

```
import random

dice1 = [1,2,3,4,5,6]
dice2 = [1,2,3,4,5,6]
a=[]
x=0
while x<10000:
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    a.append(result1 + result2)
    x=x+1
print(a)
print(len(a)) #a리스트 안의 요소의 개수를 확인할 수 있습니다
```

47. (알고리즘 10번) 문제 46번의 코드를 이용해서 두개의 주사위의 눈의 합이 10이 되는 확률을 구하시오!

```
import random

dice1 = [1,2,3,4,5,6]
dice2 = [1,2,3,4,5,6]
a=[]
x=0
while x<10000:
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10: #두개의 주사위의 눈의 합이 10이면
        a.append(result1 + result2) #a리스트에 추가해라~
    x=x+1 #x를 1씩 증가시킨다
print(len(a)/10000) #x리스트에 있는 요소를 10000으로 나눈다.
```

48. 위의 말이 맞는지 for loop문으로 테스트 해서 확인하시오!
파이썬에서는 정수형 변수가 담을 수 있는 상수의 범위가 어떻게 되는지 테스트 하시오!

```
for i in range(1000000000000000000000000000000000000000000000000):
    a=i
    print(i)
```

49. 3의 2승을 파이썬으로 구하세요!

```
print(3**2)
```

50. $\sqrt{4}$ 를 파이썬으로 구하세요!

```
import math
print(math.sqrt(4))
```

51. (알고리즘 21번) 몬테 카를로 알고리즘을 이용해서 원주율의 근사값을 파이썬으로 구하세요!

```
import random
cnt = 0
```

```
for i in range(0,10001): #10000번 루프를 돌리는데
    result1 = random.uniform(0,1) #0~1사이의 난수를 result1에 담습니다.
    result2 = random.uniform(0,1) #0~1사이의 난수를 result2에 담습니다

    if (result1**2) + (result2**2) <=1 : #부채꼴안의 점이라면
        cnt = cnt+1  #cnt를 1증가시킵니다.
print(cnt/10000*4)
```

52. 아래의 수학식을 파이썬으로 구현해서 답을 출력하시오!

$(1-2i)^2$ **-(1-2i)-12**

답 :

```
x = complex(1,-2)
result =x**2 -2*x -12
print(result) #(-17+0j)
```

53. 주사위 2개를 10번 던지시오!

```
import random

dice=[1,2,3,4,5,6]
for i in range(1,11): #아래의 실행문을 10번 수행하겠다.
    result = random.choice(dice) #실행문1
    print(result) #실행문2
```

54. 주사위를 10번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오!

```
import random

dice=[1,2,3,4,5,6]
cnt = 0

for i in range(1,11):
    result = random.choice(dice) #실행문1
    if result == 3:                    #실행문2
        cnt=cnt+1 #4칸 띄우고 들어가야 if문의 실행문이 된다.(if문의 영향권 안이 됨)
print(cnt)          #print는 변수 안에 들어 있는 값을 출력하는 함수
```

55. 주사위를 10번 던져서 주사위의 눈이 짝수가 나오는 횟수를 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
cnt = 0

for i in range(1,11):
    result = random.choice(dice)
    if result in (2,4,6):        #여러개의 값을 비교할 때 in을 사용함
```

```
        cnt = cnt+1
print(cnt)
```

56. 주사위 2개를 동시에 던져 두개의 눈의 합이 10이 되는 횟수를 출력하시오!

(주사위 2개를 20번 던지세요!)

```
import random #random이라는 모듈을 지금 이 코드에서 사용하겠다.

dice1 = [1,2,3,4,5,6]
dice2 = [1,2,3,4,5,6]
cnt = 0 # 횟수를 담기 위한 cnt라는 변수에 숫자 0을 할당한다.

for i in range(1,21): #20번 반복하기 위한 루프문
    result1 = random.choice(dice1) #실행문1
    result2 = random.choice(dice2) #실행문2
    if result1 + result2 == 10:
        cnt = cnt + 1 #if문의 영향권 안에 있는 실행문
print(cnt)
```

57. 주사위의 2개를 동시에 던져서 두 눈의 합이 10이 되는 확률을 출력하시오!

```
import random

dice1 = [1,2,3,4,5,6]
dice2 = [1,2,3,4,5,6]
cnt = 0

for i in range(1,101):
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt = cnt + 1
print(cnt/100)
```

58. 아래의 수학식을 파이썬으로 구현하시오!

$2 \times \log_2 10 + \frac{1}{3} \times \log_2 10$

```
print(2*math.log2(10) + (1/3) * math.log2(10))
```

59. 주사위 10번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오! (축약 연산자를 이용해서 출력하시오!)

```
import random

dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,11):
        result =  random.choice(dice)
```

```
        if result == 3:
            cnt += 1 #축약을 사용할 수도 있습니다.
print(cnt)
```

60. 주사를 20번 던져서 주사위의 눈이 4이상 나오는 횟수를 출력하시오!

```
import random

dice1 = [1,2,3,4,5,6]
cnt = 0

for i in range(1,21):
    result = random.choice(dice)
    if result >= 4:
        cnt = cnt +1
print(cnt)
```

61. (통계 정규분포 문제) 주사위 한개를 288회 던져서 주사위의 눈이 5이상 나오는 횟수를 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,289):
    result = random.choice(dice)
    if result >= 5:
        cnt += 1
print(cnt)
```

62. (통계 정규분포 문제) 주사위 한개를 288회 던져서 주사위의 눈이 5이상 나오는 확률을 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
cnt = 0
for i in range(1,289):
    result = random.choice(dice)
    if result >= 5:
        cnt += 1
print(cnt/288)
```

63. 주사위 한개를 288회 던질 때 5이상의 눈이 나올 횟수를 출력하는데 이 작업을 100번 해서 횟수 100개가 출력되게 하시오!

```
import random

dice = [1,2,3,4,5,6]
for j in range(1,101):
    cnt = 0
    for i in range(1,289):
        result = random.choice(dice)
        if result >= 5:
```

```
            cnt += 1
    print(cnt/288)
```

64. (점심시간 문제) 위에서 출력된 횟수 100개를 비어있는 a리스트에 담고 a리스트의 개수를 출력하시오.

a.append사용 , len(a)사용

```
import random
dice = [1,2,3,4,5,6]
a = []
for j in range(100):
    cnt = 0 #실행문1
    for i in range(1,289):                #실행문2
        result = random.choice(dice)
        if result >= 5:
            cnt += 1
    a.append(cnt)  # 반복실행, for문 안에 있음/ 실행문3
print(len(a))
```

설명 : 위의 코드는 주사위를 288번 던져서 주사위의 눈이 5이상 나올 횟수 100개를 a리스트에 입력하는 코드

65. 동전을 100번 던져서 앞면이 나오는 횟수를 출력하시오!

```
import random

coin = ['앞면', '뒷면']
cnt = 0

for i in range(1,101):
    result = random.choice(coin)
    if result == '앞면':
        cnt += 1
print(cnt)
```

66. 동전 한개와 주사위 한개를 동시에 100번 던져서 동전이 앞면이 나오고 주사위의 눈이 5가 나오는 횟수를 출력하시오!

```
import random

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
cnt = 0

for i in range(1,101):
    result1 = random.choice(coin)
    result2 = random.choice(dice)
    if result1 == '앞면' and result2 == 5:
        cnt += 1
```

```
print(cnt)
```

67. 동전한개와 주사위 한개를 동시에 100번 던져서 동전이 앞면이 나오고 주사위의 눈이 5가 나오는 횟수를 출력하는 행위를 50번해서 횟수가 50개가 출력되게 하시오!

```
import random

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result1 = random.choice(coin)
        result2 = random.choice(dice)
        if result1 == '앞면' and result2 == 5:
            cnt += 1

    print(cnt)
```

68. 문제 67번 횟수 50개를 비어있는 a 리스트에 담으시오!

```
import random

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
a = []

for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result1 = random.choice(coin)
        result2 = random.choice(dice)
        if result1 == '앞면' and result2 == 5:
            cnt += 1
    a.append(cnt)
    print(a) # 이렇게 하면 담기는 과정을 볼 수 있다.
```

69. 담겨진 a 리스트의 요소들의 개수를 확인하시오

```
import random

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
a = []

for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result1 = random.choice(coin)
```

```
        result2 = random.choice(dice)
        if result1 == '앞면' and result2 == 5:
            cnt += 1
    a.append(cnt)
print(len(a))
```

70. 위의 a리스트의 요소들의 평균값을 출력하시오

(numpy를 이용해서 구현하시오!)

↓

딥러닝에 필요한 수학 연산을 쉽게 구현하기 위해서 만들어 놓은 모듈(특정 목적을 위해 만든 코드의 집합)

Ex 1) 아래의 리스트의 요소들의 평균값을 출력하시오!

(numpy를 사용하세요)

```
import numpy as np #넘파이 모듈을 이 코드에서 사용하겠다 / 넘파이의 별칭을 np 라고 하겠다

b = [ 7,4,5,3,2 ]
print(np.mean(b))
```

Ex 2) 아래의 리스트의 요소들의 평균값을 출력하시오!

(numpy를 사용하지 않고 수행하시오)

```
import random

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
a = []

for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result1 = random.choice(coin)
        result2 = random.choice(dice)
        if result1 == '앞면' and result2 == 5:
            cnt += 1
    a.append(cnt)
print(len(a))
```

문제 70번 답:

```
import random
import numpy as np

coin = ['앞면', '뒷면']
dice = [1,2,3,4,5,6]
a = []
```

```
for j in range(1,51):
    cnt = 0
    for i in range(1,101):
        result1 = random.choice(coin)
        result2 = random.choice(dice)
        if result1 == '앞면' and result2 == 5:
            cnt += 1
    a.append(cnt)
print(np.mean(a))
```

71. 문제 64번 코드를 가져와서 a리스트의 요소들의 평균과 분산과 표준편차를 출력하시오!

```
import random
import numpy as np

dice = [1,2,3,4,5,6]
a = []
for j in range(100):
    cnt = 0 #실행문1
    for i in range(1,289):                #실행문2
        result = random.choice(dice)
        if result >= 5:
            cnt += 1
    a.append(cnt)  # 반복실행, for문 안에 이씀/ 실행문3
print(np.mean(a))
print(np.var(a))
print(np.std(a))
```

72. 위에 a리스트 담긴 요소 100개를 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
a=[]
for j in range(100):
    cnt = 0
    for i in range(1,289):
        result = random.choice(dice)
        if result >=5:
            cnt += 1
    a.append(cnt)
print(a)
>>>
```

맨 마지막에 a 리스트를 print하면 아래와 같이 a 리스트의 모든 요소들을 한번에 볼 수 있다.

[102, 110, 97, 111, 88, 100, 92, 95, 91, 97, 108, 102, 112, 95, 90, 90, 97, 110, 94, 95, 92, 107, 88, 91, 84, 106, 100, 93, 85, 92, 96, 107, 94, 97, 107, 102, 93, 109, 107, 90,

93, 90, 82, 96, 103, 96, 93, 83, 101, 105, 95, 83, 95, 99, 98, 102, 96, 105, 92, 96, 90, 103, 91, 110, 97, 91, 99, 91, 86, 91, 101, 97, 91, 95, 79, 115, 106, 96, 105, 103, 96, 101, 95, 100, 103, 102, 104, 106, 90, 91, 93, 93, 89, 91, 99, 87, 99, 102, 102, 103]

73. 위의 a리스트의 요소들을 하나씩 빼내서 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
a=[]
for j in range(100):
    cnt = 0
    for i in range(1,289):
        result = random.choice(dice)
        if result >=5:
            cnt += 1
    a.append(cnt)

for k in a:
    print(k)
```

74. 그러면 a 리스트의 요소들의 숫자가 90 이상이고 106이하인 요소들의 개수를 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
a=[]
for j in range(100):
    cnt = 0
    for i in range(1,289):
        result = random.choice(dice)
        if result >=5:
            cnt += 1
    a.append(cnt)

cnt2 = 0
for k in a:
    if k>= 90 and k<=106:          # 90<= k <= 106 이렇게 해도 됨
        cnt2= cnt2 + 1

print(cnt2)
```

75. 그러면 a 리스트의 요소들의 숫자가 90 이상이고 106이하인 요소들의 확률을 출력하시오!

```
import random

dice = [1,2,3,4,5,6]
a=[]
for j in range(100):
    cnt = 0
    for i in range(1,289):
```

```
        result = random.choice(dice)
        if result >=5:
            cnt += 1
    a.append(cnt)

cnt2 = 0
for k in a:
    if 90<= k <=106:
        cnt2= cnt2 + 1
print(cnt2/100)
```

76. (SQL ---> 판다스 문법) 사원 테이블에서 이름과 월급을 출력하시오.

```
SQL > select ename, sal
            from emp;
```

판다스 >

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['ename', 'sal']])
```

설명 : 데이터프레임명 [['컬럼명1', '컬럼명2']]

77. 아래의 SQL을 판다스로 구현하시오!

```
select ename, sal, job, hiredate
      from emp;
```

답:

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename', 'sal', 'job', 'hiredate']])
```

78. 아래의 SQL을 판다스로 구현하시오!

```
select ename, sal
      from emp
      where job = 'SALESMAN';
```

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emmp3.csv")
print(emp[['ename', 'sal']][검색조건])
>>>
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['job']=='SALESMAN'])
```

79. 월급이 3000이상인 사원들의 이름과 월급을 출력하시오!

```
SQL > select ename, sal
            from emp
            where sal >= 3000
```

판다스>

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['sal']>=3000])
```

80. 아래의 SQL을 판다스로 구현하시오!

```
SQL > select ename, sal
            from emp
            where sal between 1000 and 3000;
```

판다스>

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['sal'].between(1000,3000)])
```

81. 월급이 1000에서 3000사이가 아닌 사원들의 이름과 월급이 출력하시오!

(not이 판다스에서는 ~입니다.)

```
SQL > select ename, sal
            from emp
            where sal not between 1000 and 3000
```

판다스:

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename', 'sal']][~emp['sal'].between(1000,3000)])
```

82. 직업이 CLERK, SALESMAN인 사원들의 이름과 직업을 출력하시오!

```
SQL > select ename, job
            from emp
            where job in ('CLERK','SALESMAN');
```

판다스 >

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename','sal']][emp['job'].isin(['CLERK','SALESMAN'])])
```

83. 이번에는 직업이 CLERK, SALESMAN이 아닌 사원들의 이름과 직업을 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename','sal']][~emp['job'].isin(['CLERK','SALESMAN'])])
```

84. 커미션이 null인 사원들의 이름과 커미션을 출력하시오!

```
SQL >
select ename, comm
      from emp
      where comm is null
```

판다스 >

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
```

```
print(emp[['ename','comm']][emp['comm'].isnull()])
```

85. 커미션이 null이 아닌 사원들의 이름과 커미션을 출력하시오!

SQL >

```
select ename, comm
        from emp
        where comm is not null
```

판다스 >

```
import pandas as pd
emp = pd.read_csv("c:₩₩data1₩₩emp3.csv")
print(emp[['ename','comm']][~emp['comm'].isnull()])
```

(오늘의 마지막문제) 확률문제

6개의 제품이 들어있는 상자가 있는데 그 중에 2개가 불량품이라고 하자. 제품검사를 위해서 3개를 추출햇을 때 적어도 한개가 불량품일 확률은?
(수학식으로 하지 말고 파이썬으로 아래와 같이 상자안에 제품 6개를 넣고 랜덤으로 추출해서 수행하시오! (복원 추출로 하세요. 한번 추출한 제품은 다시넣지 않는다)

box = ['정상','정상', '불량', '정상', '불량', '정상']

```
import random

box = ['정상','정상', '불량','정상', '불량', '정상']
for I in range(1,4):
    cnt = 0
    result = random.choice(box)
    if result == '정상':
        cnt +=1
    else:
        continue
print(1-(cnt/3))
```

86. 이름이 SCOTT인 사원의 이름과 월급을 출력하시오

SQL >

```
 select ename, sal
            from emp
```

Pandas >

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal'] ][emp['ename'] == 'SCOTT'])
```

emp[컬럼 리스트] [조건]

87. 아래의 scott을 담은 문자형 변수 a에서 알파벳 o를 출력하시오~

a = 'scott'

print(a) #scott

답 :
a = 'scott'
print(a[2])

설명 : 문자형 변수에서 특정 철자만 가져오고 싶으면 문자형변수 옆에 대괄호를 쓰고 순서에 해당하는 숫자를 적어주면 됩니다.

88. 아래의 문자형 변수에서 맨 끝의 철자인 h를 출력하시오!
b = 'smith'

답:
b= 'smith'
print(b[4])

또는
b= 'smith'
print(b[-1])

	s	m	i	t	h
자리수	0	1	2	3	4
	-5	-4	-3	-2	-1

89. 판다스를 이용해서 emp3.csv에서 이름을 출력하는데 이름의 첫번째 철자만 출력하시오!

SQL >
select substr(ename,1,1)
from emp

import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['ename'])

↓

import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv") #emp3.csv 에서 ename만 가져와서
for i in emp['ename']: #ename의 개수만큼 loop문을 수행해서
print(i[0]) #ename의 첫글자를 출력합니다.

import pandas as pd

```
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    print(type(i))
```

이렇게 하면

```
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
```

이렇게 나오는데 이게 문자열이라는 뜻이다.

문자열만 잘라낼 수 있다.

```
print(emp['ename'])
print(type(emp['ename']) # <class 'pandas.core.series.Series'>
```

설명 : type명령어는 변수가 문자형, 숫자형, 리스트형, 딕셔너리형, 튜플형인지 확인하는 함수 입니다.

```
a= 'scott'
print(type(a)) #<class 'str'>
print(a[0])
```

설명 : <class 'str'>문자형으로 변경해줘야 문자형에서 특정요소를 가져오는 문법을 사용할 수 있습니다.

Ex) a[0]

판다스의 <class 'pandas.core.series.Series'>를 문자형으로 변경하려면 for loop 문을 이용해서 하나씩 가져오면 됩니다.

Ex) for i in emp['ename']:
        print(i)

90. 판다스를 이용해서 emp3.csv를 가져와서 이름의 끝글자를 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    print(i[-1])
```

91. 위의 결과에서 앞에 이름도 같이 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
        print(i, ' ', i[-1])
```

92. 이름의 첫번째 철자가 S로 시작하는 사원들의 이름을 출력하시오!

SQL >

```
select ename
        from emp
        where ename like 'S%':
```

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    if i[0] == 'S':
        print(i)
```

93. 이름의 끝글자가 T로 끝나는 사원들의 이름을 출력하시오!

SQL>

```
select ename
        from emp
        where ename like '%T'
```

Pandas >

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    if i[-1] == 'T':
        print(i)
```

94. 이름의 두번째 철자가 M인 사원의 이름을 출력하시오

SQL >

```
select ename
        from emp
        where ename like '_M%'
```

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
        if i[1] == 'M':
                print(i)
```

95. 아래의 SQL을 판다스로 구현하시오!

SQL > select substr(ename, 1, 3)
from emp;

답 :

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    print(i[0:3])
```

96. 아래의 SQL을 판다스로 구현하시오!

SQL>
select substr(ename, -2, 2)
from emp

ex) K I N G
-4 -3 -2 -1
>> NG

답:

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    print( i[-2 :  ] )
```

↑
콜론(:) 다음에 아무것도 안넣으면 그냥 끝까지 읽어라 라는 뜻입니다.

97. (점심시간 문제) 아래의 SQL을 판다스로 구현하시오!

SQL >
select substr(ename, 3, 2)
from emp;

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
    print(i[2:4])
```

98. 아래의 두개의 리스트를 연결하여 출력하세요.

a = [2,3,4,5]
b = [9,2,4,8]

답 :

```
a = [2,3,4,5]
```

```
b = [9,2,4,8]
print(a+b)
```

99. 아래의 SQL을 판다스로 구현하시오!

SQL > select ename || sal
from emp;

```
Pandas>
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i, k in zip(emp['ename'], emp['sal']):
    print(i + str(k))
```

설명 : zip함수를 for loop문에 사용하게 되면 두개의 범위 데이터를 한번에 받아서 loop를 수행할 수 있습니다.
str함수를 사용한 이유는 문자형 + 문자형은 연결이 가능한데 문자형 + 숫자형은 연결이 안됩니다. 그래서 숫자형을 문자형으로 변환하기 위해서 str함수를 사용했습니다.

100. 주사위의 눈 6개를 100개 담은 리스트 dice100을 만들고 dice100 리스트를 출력하시오!

```
dice100 = [1,2,3,4,5,6] * 100
print(dice100)
```

101. 초등학생 10명의 키가 들어있는 아래의 tall리스트의 요소 10개를 10000개로 증가시켜서 tall10000 리스트에 담고 출력하시오!
tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]

답 :

```
tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
print(tall10000)
```

102. 위의 모집단의 평균값, 분산, 표준편차를 출력하시오!

```
import numpy as np

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
print(tall10000)


print(np.mean(tall10000)) #136.66 모집단의 평균
print(np.var(tall10000)) #23.782399999999985 모집단의 분산
print(np.std(tall10000)) #4.876720209321013 모집단의 표준편차
```

설명 : np.mean(리스트)를 사용하면 평균값이 출력됩니다.

103. 위의 모집단 tall10000에서 표본을 20개를 랜덤으로 추출하시오!

```
import random

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]

tall10000 = tall * 10000 #초등학생 키 데이터 10000개 생성(모집단 생성)
for i in range(1,21): #20번 반복 하면서
    print(random.choice(tall10000)) #모집단에서 랜덤으로 키 하나를 추출
```

104. 위의 랜덤으로 추출한 표본 20개를 비어있는 a리스트에 담으시오!

```
import random
import numpy as np

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
a = [] #표본을 담기 위해 a라는 비어있는 리스트를 만듭니다.

for i in range(1,21): #20번 반복 하면서
    a.append(random.choice(tall10000)) #모집단에서 랜덤으로 키 하나를 추출
                                       #추출해서 a리스트에 담는다.
```

105. 위의 표본 20개의 평균값을 출력하시오!

```
import random
import numpy as np

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
a = [] #표본을 담기 위해 a라는 비어있는 리스트를 만듭니다.

for i in range(1,21): #20번 반복 하면서
    a.append(random.choice(tall10000)) #모집단에서 랜덤으로 키 하나를 추출
                                       #추출해서 a리스트에 담는다.
print(np.mean(a))
```

106. 아래의 SQL을 판다스로 구현하시오!

```
SQL > select ename, length(ename)
            from emp;
```

답:

```
Pandas>
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
for i in emp['ename']:
      print(i, len(i))
```

107. 모평균이 30이고 모표준편차가 7인 모집단을 구성하시오!

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
print(mogipdan)
```

설명 : np.random.randn(숫자)를 쓰면 이 숫자만큼 가우시안 정규분포에 따르는 난수들이 숫자만큼 생성됩니다.

```
[22.91933477 22.47405201 21.33636965 ... 27.25153451 32.53801239
 27.14394364]
```

108. 문제 107번의 모집단의 모평균을 출력하시오!

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
print(np.mean(mogipdan))                    #30.00285676335108
```

109. 아래의 SQL을 판다스로 구현하시오!

```
SQL > select ename, sal
            from emp
            where job in ('SALESMAN', 'ANALYST')
```

답 :

```
pandas>

import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print( emp [ [ 'ename', 'job' ] ] [ emp[ 'job' ].isin( [ 'SALESMAN','ANALYST' ] ) ] )
```

110. 아래의 SQL을 판다스로 구현하시오!

```
SQL > select ename, job
            from emp
            where job not in ('SALESMAN', 'ANALYST')
```

답 :

```
pandas>

import pandas as pd
```

```
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print( emp [ [ 'ename', 'job' ] ] [ ~emp[ 'job' ].isin( [ 'SALESMAN','ANALYST' ] ) ] )
```

111. 아래와 같은 글씨가 출력되게 하시오!

"모집단"의 모평균은

"모집단"의 모분산은

"모집단"의 모표준편차는

답 :

```
a = ' "모집단"의 모평균은'
b = '"모집단"의 모분산은'
c =' "모집단"의 모표준편차는'
print(a, b, c)
```

또는

```
a = """ "모집단"의 모평균은"""
b = """ "모집단"의 모분산은"""
c ="""  "모집단"의 모표준편차는 """
print(a, b, c)
```

112. 모평균이 30이고 모표준편차가 7인 모집단 1000000개의 모평균과 모분산과 모표준편차를 아래와 같이 출력하시오!

"모집단"의 모평균은 30.000019293

"모집단"의 모분산은 49.00192922

"모집단"의 모표준편차는 7.00191929

답:

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg

a = ' "모집단"의 모평균은'
b = '"모집단"의 모분산은'
c =' "모집단"의 모표준편차는'

print(a, np.mean(mogipdan))
print(b, np.var(mogipdan))
print(c, np.std(mogipdan))
```

113. 모평균이 30이고 모표준편차가 7인 모집단(1000000개)에서 표본을 49를 뽑으시오

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
print( np.random.choice( mogipdan, 49 ) )

#
[27.33861713 20.22688007 33.27040213 20.84062477 35.5745421  36.60337036
 31.06303694 27.60667511 39.85748886 37.37385465 32.4787406  30.31904379
 41.20048302 31.5496449  38.94813071 33.02402016 35.60627332 30.15454711
 37.71209098 37.8043009  23.56524569 23.59512753 26.42617715 40.5546099
 34.96515401 30.48858959 34.00003914 29.68916875 32.53056455 46.17221216
 15.17356597 33.08688253 38.64063573 18.71812957 28.49894548 44.81922873
 26.50984959 37.97540133 42.73306192 16.93228218  7.16919842 29.95596001
 31.45662218 32.36413341 32.79543525 43.74556115 32.40160423 28.29380756
 26.18524689]
```

114. 문제 113에서 뽑은 49개의 평균값을 출력하시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
print(np.mean(np.random.choice( mogipdan, 49)))
```

또는

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
print(np.random.choice( mogipdan, 49).mean ()) #29.123034288802877
```

설명 : numpy의 random안의 choice함수를 이용해서 mogipdan에서 49개를 표본샘플링하고 그 표본의 평균값을 출력한다.

115. 문제 114에서 출력한 표본의 평균값을 하나가 아니라 100개가 출력되게 하시오!

```
import numpy as np

avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg

for i in range(1,101):
    print(np.random.choice( mogipdan, 49).mean ()) #29.123034288802877
```

116. 이번에는 100개를 출력하지 말고 비어있는 리스트 a를 선언아고 a에 100개를 담으시오!

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

for i in range(1,101):
    a.append(np.random.choice(mogipdan, 49).mean ())
print(a)
```

117. 문제 116번에서 구한 표본평균의 평균값과 표준편차를 아래와 같이 출력하시오!

"표본평균"의 평균값은 ?

"표본평균"의 표준편차는 ?

답:

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

x = '"표본평균"의 평균값은'
y =  '"표본평균"의 표준편차는'


for i in range(1,101):
    a.append(np.random.choice(mogipdan, 49).mean ())
print(x, np.mean(a))
print(y,np.std(a))
```

아니면 x, y값 지정하지 않고

print(""" "표준평균"의 평균값은 """, np.mean(a))

print(""" "표준평균"의 표준편차는 """,np.std(a)) 이렇게 해준다.

118. 아래의 변수를 이용해서 아래와 같이 결과가 출력되게 하시오

num1 = 5
num2 = 10

결과 : 5는 10보다 작습니다.

답:

```
num1 = 5
num2 = 10
print('%d는 %d보다 작습니다' %(num1, num2))
```

119. 문제117번의 결과가 아래와 같이 출력되게 하시오!

(문자열 포멧을 이용하세요)

표본평균의 평균값은 ? 이고 분산값은 ? 이고 표준편차는 ? 입니다.

답:

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

for i in range(1,101):
    a.append(np.random.choice(mogipdan, 49).mean ())
print('표본평균의 평균값은 %f이고 분산값은 %f이고 표준편차 값은 %f입니다'
    %(np.mean(a), np.var(a), np.std(a)))
```

120. 위의 결과가 아래와 같이 소수점 두번째 자리까지만 나오게 하시오!

표준평균의 평균값은 30.06이고 분산값은 0.78이고 표준편차 값은 0.89입니다'

```
import numpy as np

avg = 30
std = 7
N = 1000000

mogipdan = np.random.randn(N) * std + avg
a = []

for i in range(1,101):
    a.append(np.random.choice(mogipdan, 49).mean ())
print('표본평균의 평균값은 %.2f이고 분산값은 %.2f이고 표준편차 값은 %.2f입니다'
%(round(np.mean(a),2), round(np.var(a),2), round(np.std(a),2) ))
```

121. (통계를 코드로 구현 문제4번의 링크2) 오늘의 마지막 문제

```
import numpy as np
import random

avg = 18
std = 3
N = 1000000
a = []

mgroup = np.random.randn(N) * std + avg

for i in range(1,10001):
    result = np.random.choice(mgroup, 36).mean ()
    if result >= 17:
        a.append(result)
print(len(a)/10000)
```

```
import numpy as np
import random


avg = 18
std = 3
N = 1000000

mgroup = np.random.randn(N) * std + avg
a = []
u=17  # 짐의 평균무게 (표본평균)

for i in range(1,101):
    a.append(np.random.choice(mgroup, 36).mean ())
z =((u-round(np.mean(a),2))/round(np.std(a),2) )
if -2 <= z <= 0:
    z = 0.4772
    print(z + 0.5)
```

122. 동전의 앞면과 뒷면을 포함하는 리스트 coin을 만드시오!

```
coin = ['앞면', '뒷면']
```

123. 문제 122번에서 만든 coin의 요소를 10000개로 늘려서 coin_10000 변수에 담으시오!

```
coin = ['앞면', '뒷면']
coin_10000 = coin*10000
print(coin_10000)
```

124. 위에서 만든 coin_10000 리스트에서 표본을 10개를 추출하시오!

```
import numpy as np
```

```
coin = ['앞면', '뒷면']
coin_10000 = coin*10000
print(np.random.choice(coin_10000,10))
```

설명 : np.random.choice은 numpy모듈안에 random코드 안에 choice 함수를 사용하겠다. np.random.choice(모집단리스트, 샘플 개수)

결과 :

['뒷면' '뒷면' '앞면' '뒷면' '뒷면' '뒷면' '앞면' '앞면' '앞면' '앞면']

125. 문제 124번에서 추출한 샘플 10개에서 앞면이 몇번 나오는지 출력하시오!

```
import numpy as np

coin = ['앞면', '뒷면']
coin_10000 = coin*10000
result = np.random.choice(coin_10000,10)
cnt = 0
for i in result: #['뒷면' '뒷면' '앞면' '뒷면' '뒷면' '뒷면' '앞면' '앞면' '앞면' '앞면']
    if i == '앞면':
        cnt = cnt + 1
print(cnt)
```

126. a리스트의 인덱스 번호 3번째 요소를 17로 변경하시오

```
a = [1, 2, 3, 4, 5]
      0  1  2  3  4
```

답 :

```
a = [1, 2, 3, 4, 5]
a[3] = 17
print(a)
```

127. 아래의 숫자 데이터들을 튜플로 생성하시오!

튜플 변수 이름은 point라고 하세요!

0.01, 0.02, 0.03, 0.04, 0.05

```
point = ( 0.01, 0.02, 0.03, 0.04, 0.05)
print( type(point) ) #<class 'tuple'>
```

128. 문제 128번의 튜플 point의 요소 중 0.03만 뽑아서 출력하시오!

```
point = ( 0.01, 0.02, 0.03, 0.04, 0.05)
print( point[2] )
```

129. 문제 128번의 튜플의 모든 요소를 다 출력하시오!

결과:

0.01
0.02
0.03
0.04
0.05

답:

```
point = ( 0.01, 0.02, 0.03, 0.04, 0.05)
for i in point: #하나씩 빼낸다. point에서
    print(i)
```

번외)

```
for j in 'scott':
      print(j)
```

이것도 된다. s c o t t 하나씩 나온다.

130. 아래의 두개의 리스트를 각각 만들고 아래와 같이 fruit라는 딕셔너리를 생성하시오! (for문에 zip사용하면 편하게 할 수 있음)

a = ['사과', '배', '포도', '복숭아', '바나나']
b = ['apple', 'pear', 'grape', 'peach', 'banana']

결과:
{'사과': 'apple', '배': 'pear', '포도': 'grape', '복숭아': 'peach', '바나나': 'banana'}

답:

```
a = ['사과', '배', '포도', '복숭아', '바나나']
b = ['apple', 'pear', 'grape', 'peach', 'banana']
fruit = { } #비어있는 딕셔너리를 생성한다.

for i, k in zip(a,b):
    print(i,k)
```

사과 apple
배 pear
포도 grape
복숭아 peach
바나나 banana

131. (점심시간 문제) 아래의 SQL을 파이썬으로 구현하시오!

```
SQL > select lower(ename), lower(job)
            from emp;
```

답:

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
i = emp['ename']
j = emp['job']

for a, b in zip(i,j):
    print(a.lower(),b.lower())
```

132. for loop문을 사용하여 1부터 10까지 출력하시오!

```
for i in range(1,11):
    print(i)
```

133. 1부터 10까지 다 더한 값을 출력하시오!

```
cnt = 0
for i in range(1,11):
    cnt = cnt + i
print(cnt)
```

134. 위의 코드를 이용해서 함수를 생성하는데 아래와 같이 숫자를 입력하고 함수를 실행하면 해당 숫자까지 1부터 다 더한 값이 출력되게 하시오!
print(all_add(10))

답:

```
def all_add(n1):
        cnt = 0
        for i in range(1,n1+1):
            cnt = cnt + i
        return cnt                #함수 생성시 retrun절은 필수 입니다.

print(all_add(10))
```

135. 다음의 문자열 변수를 생성하고 문자열 변수의 문자를 소문자로 출력하시오!

```
a = 'SCOTT'
```

답:

```
a = 'SCOTT'
print(a.lower( ))
```

또는

```
print('SCOTT'.lower( ))
```

136. 아래의 문자열을 대문자로 출력하시오!

```
a = 'scott'
```

답:

```
a = 'scott'
```

```
print(a.upper( ))

또는
print('scott', upper( ))
```

137. 아래의 문자열에서 첫번째 철자만 출력하시오!
a = 'scott'

답:
```
a = 'scott'
print(a[0])
```

138. 문제 137번에서 출력한 첫번째 철자를 대문자로 출력하시오!
```
a = 'scott'
print(a[0].upper())
```

139. 아래의 문자열 변수에서 cott만 출력하시오
(두번째 철자부터 끝까지 출력하시오)
a = 'scott'

답:
```
print(a[1:])
```

140. 아래의 함수를 생성하시오.
print(initcap('scott')
결과: Scott

답:
```
def initcap(val):
    return val[0].upper() + val[1:].lower()

print(initcap('scott'))
```

141. abs함수를 사용하지 말고 if문을 이용해서 절대값을 출력하는 my_abs라는 함수를 생성하시오!
print(my_abs(-9)) #9
print(my_abs(9)) #9

답:
```
def my_abs(num1):
    if num1 >0:            # 0보다 크면
        return num1       # 입력값 그대로 리턴
    else:                    # 그게 아니라면 (0에 음수를 써도 0은 0)
        return -num1     # 음수를 붙여 리턴
```

```
print( my_abs(-5) ) #5
```

142. 서울시 초등학생 백만명의 키를 모집단으로 구성하는데 평균키가 148.5이고 표준편차가 30인 모집단을 만드시오!

```
import numpy as np  # numpy 모듈을 임폴트 합니다.

avg = 148.5
std = 30
N = 1000000
mogip = np.random.randn(N) * std + avg
print(mogip)
```

143. 위의 모집단에서 100명을 표본으로 추출하여 100명의 평균키를 비어있는 리스트 a에 입력하는 작업을 10000번 수행하여 a 리스트에 10000개의 표본의 평균키가 입력되게 하시오!

```
import numpy as np

avg = 148.5
std = 30
N = 1000000
mogip = np.random.randn(N) * std + avg

a = []
for i in range(1,10001):
    a.append(np.random.choice(mogip, 100).mean())
print(a)
```

설명 : 초등학생 100만명의 키 모집단에서 표본을 100개 추출해서 표본 평균을 10000개를 모았다

144. 통계학을 전문으로 구현하는 모듈인 scipy 모듈을 임폴트하여 위의 표본의 평균키값에 대한 확률밀도값을 출력하시오!

```
import numpy as np
from scipy.stats import norm      #scipy의 stats 패키지로부터 norm이라는 모듈을 import해라

avg = 148.5
std = 30
N = 1000000

height = np.random.randn(N) * std + avg
a = []

for i in range(1,10001):
    result = np.random.choice(height, 100).mean()    #모집단에서 100개 추출
    a.append(result)        #평균값을 10000번 입력함
```

```
x = np.arange(140, 160, 0.001)        #140~160까지 0.001간격으로 숫자를 생성
y = norm.pdf(x,표본평균값들의 평균값, 표본평균값들의 표준편차)

#초등학생 키의 표본평균값들에 대한 확률 밀도함수 값이 출력됩니다.
print(y)
```

```
>>>
import numpy as np
from scipy.stats import norm #scipy의 stats 패키지로부터 norm이라는 모듈을 import해라
                                    # from 패키지 import 모듈이름 --> 모듈을 다 모은게 패키지
                                    # norm을 쓸 때는 꼭 from spicy.stats를 써줘야한다.

avg = 148.5
std = 30
N = 1000000

height = np.random.randn(N) * std + avg
a = []

for i in range(1,10001):
    result = np.random.choice(height, 100).mean() #모집단에서 100개 추출
    a.append(result) #평균값을 10000번 입력함

x = np.arange(140, 160, 0.001) #140~160까지 0.001간격으로 숫자를 생성
y = norm.pdf(x,np.mean(a), np.std(a))

print (y)
```

145. 데이터 시각화 전문 모듈인 matplotlib를 임폴트하여 143번 문제의 표본 평균값 10000개에 대한 확률밀도함수 값으로 정규분포 그래프를 그리시오~

```
import numpy as np
from scipy.stats import norm #scipy의 stats 패키지로부터 norm이라는 모듈을 import해라
import matplotlib.pyplot as plt #matplotlib 모듈안에 pyplot라는 함수를 import하는데 별칭을 plt로 해라

avg = 148.5
std = 30
N = 1000000

height = np.random.randn(N) * std + avg
a = []
```

```
for i in range(1,10001):
    result = np.random.choice(height, 100).mean() #모집단에서 100개 추출
    a.append(result) #평균값을 10000번 입력함

x = np.arange(140, 160, 0.001) #140~160까지 0.001간격으로 숫자를 생성
y = norm.pdf(x,np.mean(a), np.std(a))

plt.plot(x,y, color = 'blue')
plt.show()
```

결과 :

146. 문제 145번의 확률밀도함수 그래프의 아래쪽 영역도 색깔로 채우시오~

```
import numpy as np
from scipy.stats import norm #scipy의 stats 패키지로부터 norm이라는 모듈을 import해라
import matplotlib.pyplot as plt #matplotlib 모듈안에 pyplot라는 함수를 import하는데 별칭을 plt로 해라

avg = 148.5
std = 30
N = 1000000

height = np.random.randn(N) * std + avg
a = []

for i in range(1,10001):
    result = np.random.choice(height, 100).mean() #모집단에서 100개 추출
    a.append(result) #평균값을 10000번 입력함

x = np.arange(140, 160, 0.001) #140~160까지 0.001간격으로 숫자를 생성
y = norm.pdf(x,np.mean(a), np.std(a))

plt.plot(x,y, color = 'blue')
plt.fill_between9x,y, interpolate = True, color='skyblue', alpha=0.7)
```

# 설명: plt모듈안의 fill_between 함수를 이용해서 확률밀도함수 그래프의 아래 영역을 색깔로 채우는데 interpolate = True를 이용해서 아래의 영역이 색깔로 채워지게 되고 alpha는 색깔 투명도인데 0.0~1.0사이로 기술 할수 있습니다. 0.7이 예쁘게나옴^_^
plt.show()

147. **(오늘의 마지막 문제)**

문제 146번에서 그린 확률밀도함수 그래프의 색깔을 변경하고 그래프를 사진으로 첨부해서 올리세요~

```
import numpy as np
from scipy.stats import norm #scipy의 stats 패키지로부터 norm이라는 모듈을 import해라
import matplotlib.pyplot as plt #matplotlib 모듈안에 pyplot라는 함수를 import하는데 별칭을 plt로 해라

avg = 148.5
std = 30
N = 1000000

height = np.random.randn(N) * std + avg
a = []

for i in range(1,10001):
    result = np.random.choice(height, 100).mean() #모집단에서 100개 추출
    a.append(result) #평균값을 10000번 입력함

x = np.arange(140, 160, 0.001) #140~160까지 0.001간격으로 숫자를 생성
y = norm.pdf(x,np.mean(a), np.std(a))

plt.plot(x,y, color = 'r')
plt.fill_between(x,y, interpolate = True, color='lemonchiffon', alpha=0.7)
plt.show()
```

오늘의 문제(148-300)

2020년 12월 2일 수요일 오전 10:08

148. my_cal.py 모듈 스크립트 안에 곱하기를 하는 아래의 함수를 추가하시오!

```
def gob_number(n1,n2):
    result = n1 * n2
    return result
```

149. 다른 새로운 창에서 my_cal 모듈을 임포트하고 gob_number함수를 실행하시오~

```
import my_cal

print(my_cal.add_number(1,2))
print(my_cal.gob_number(1,2))
```

설명 : 그동안은 통계문제를 풀려고 import random을 했는데 random은 여러개의 함수를 모아놓은 소스코드였고 아마도 어딘가에 random.py라는 이름으로 저장되어 있는것을 우리가 import 한 것 입니다.

150. 이번에는 나누기를 하는 함수를 my_cal.py에 저장하고 다른 새로운 창에서 아래와 같이 import하고 실행될 수 있도록 하시오

```
#1
def devide(n1,n2):
    result = n1/n2
    return result

#2
import my_cal

print(my_cal.devide(10,2))
```

151. my_loc 폴더를 site-packages 폴더 밑에 두세요!

152. 아래의 행렬의 합을 출력하시오!

(1,2) + (3 1) =
(4,5) (6 2)

답:

```
import numpy as np
a = [ [ 1,2], [4,5] ] #a리스트를 numpy배열로 구성함
b = [[3,1], [6,2]]
a2 = np.array(a)
b2 = np.array(b)
print(a2 + b2)
```

결과:

```
[[ 4  3]
 [10  7]]
```

153. 아래의 행렬의 합을 출력하시오!

(6 3 4) + (4 5 7) = ?
(5 1 7) (9 20 4)

```
import numpy as np
a = [ [6,3,4], [5,1,7] ] #a리스트를 numpy배열로 구성함
b = [[4,5,7], [9,20,4]]
a2 = np.array(a)
b2 = np.array(b)
print(a2 + b2)
```

154. 아래의 행렬의 곱을 먼저 손으로 구하시오!

(1 2) x (2 1) = ?
(4 3) (3 4)

(1*2 + 2*3 1*1 + 2*4) = (8 9)
(4*2 + 3*3 4*1 + 3*4) (17 16)

답:

```
import numpy as np
a = [ [1,2], [4,3] ]
b = [[2,1], [3,4]]
a2 = np.array(a)
b2 = np.array(b)
print(np.dot(a2,b2))
```

결과:

```
[[ 8  9]
 [17 16]]
```

155. (점심시간 문제) 아래의 행렬의 곱을 출력하시오!

(3 4 1) x (2 1) = ?
(2 4 3) (4 3)
(6 7)

답:

```
a = [ [3,4,1], [2,4,3] ] #a리스트를 numpy배열로 구성함
b = [[2,1], [3,4], [6,7]]
a2 = np.array(a)
b2 = np.array(b)
print(np.dot(a2,b2))
```

결과:

```
[[24 26]
 [34 39]]
```

156. 폐사진을 파이썬에서 시각화 하시오!

```
import PIL.Image as pilimg #이미지를 파이썬에서 시각화 하기 위한 모듈
import numpy as np
import matplotlib.pyplot as plt # 데이터 시각화 전물 모듈을 임포트 합니다.

im = pilimg.open('c:₩₩data₩₩1.png') #lena.png 파일을 읽어서 im에 입력
pix = np.array(im) #numpy 배열로 변환 합니다.
plt.imshow(pix)     # 화면에 띄웁니다.

im = pilimg.open('c:₩₩data₩₩2.png') #2.png 파일을 읽어서 im에 입력
pix = np.array(im) #numpy 배열로 변환 합니다.
print(pix)
plt.imshow(pix)     # 화면에 띄웁니다.
```

print(pix) 하면

```
[[[ 33  33  33]
  [ 25  25  25]
  [ 23  23  23]
  ...
  [  0   0   0]
  [  0   0   0]
  [  1   1   1]]

 [[ 33  33  33]
  [ 25  25  25]
  [ 22  22  22]
  ...
  [  0   0   0]
  [  0   0   0]
  [  2   2   2]]

 [[ 30  30  30]
  [ 21  21  21]
  [ 18  18  18]
  ...
  [  4   4   4]
  [  1   1   1]
  [  0   0   0]]

 ...

 [[130 130 130]
  [126 126 126]
  [116 116 116]
  ...
  [ 47  47  47]
  [ 45  45  45]
  [ 44  44  44]]

 [[130 130 130]
  [125 125 125]
```

```
 [118 118 118]
 ...
 [ 54  54  54]
 [ 51  51  51]
 [ 49  49  49]]

[[161 161 161]
 [188 188 188]
 [183 183 183]
 ...
 [ 63  63  63]
 [ 62  62  62]
 [ 65  65  65]]]
```

이렇게 숫자로 뜸

157. 총설계도를 수정해서 총알을 아래와 같이 충전하면 몇발이 충전되었습니다. 라는 메세지가 출력되게 하시오.

```
gun1.charge(10)
10발이 충전되었습니다.
```

답:

```
class Gun():
    def charge(self,num):
        self.bullet = num
        print(num,'발이 충전되었습니다.')

    def shoot(self,num):
        for i in range(num):
            if self.bullet>0:
                print('탕')
                self.bullet -=1
            elif self.bullet ==0:
                print('총알이 없습니다')
                break
```

158. 이번에는 총을 쏘면 총알이 탕! 탕! 하면서 아래쪽에 몇발 남았습니다. 라는 메세지가 출력되게 하시오!

```
gun1 = Gun()
gun1.charge(10)

gun1.shoot(3)
탕!
탕!
탕!
```

7발 남았습니다.

답:

```
class Gun():
    def charge(self,num):
        self.bullet = num
        print(num,'발이 충전되었습니다.')

    def shoot(self,num):
        for i in range(num):
            if self.bullet>0:
                print('탕')
                self.bullet -=1
            elif self.bullet ==0:
                print('총알이 없습니다')
                break
        print('총알이', self.bullet, '발 남았습니다')
```

159. 총을 처음 생산했을때 총알이 반드시 0발 장전되게끔 총 설계도를 수정하시오!

```
class Gun():
    def __init__(self): #설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
        self.bullet = 0 #만들때만 작동하고 그 다음엔 작동 안함
        print('총이 만들어졌습니다', self.bullet, '발 장전 되었습니다')

    def charge(self,num):
        self.bullet = num
        print(num,'발이 충전되었습니다.')

    def shoot(self,num):
        for i in range(num):
            if self.bullet>0:
                print('탕')
                self.bullet -=1
            elif self.bullet ==0:
                print('총알이 없습니다')
                break
        print('총알이', self.bullet, '발 남았습니다')
```

160. 총클래스를 이용해서 아래와 같이 카드 클래스를 만들고 아래와 같이 카드를 충전하고 사용하시오!

```
class  Card( ): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.
    def __init__(self): # 설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
        self.cash = 0  # 만들때만 작동하고 그 다음엔 작동안함
        print ( '카드가 만들어졌습니다' , self.cash, '원이 충전 되었습니다')

    def  charge( self, num ): # 총알을 충전하는 함수
        self.cash = num
        print ( num, '원이 충전되었습니다')

    def  consume( self, num):  # 총을 쏘는 함수
```

```
        if  self.cash > 0:
            self.cash -= num
            print (num ,'원 사용되었습니다')
            print ( '잔액이  ', self.cash, ' 원 남았습니다')

        elif  self.cash ==0:
            print('잔액이 없습니다')

card1=Card()
card1.charge(10000)
card1.consume(1000)
```

161. 초등학생 키에 대한 모집단을 생성하세요! (천만개로 구성)

키의 평균값은 140, 표준편차 5로 해서 생성하시오!

```
import numpy as np

height = np.random.randn(10000000) * 5 + 140
print(height)
```

162. 위의 모집단에서 표본을 100개를 추출해서 표본의 평균값을 출력하시오

```
import numpy as np

height = np.random.randn(10000000) * 5 + 140
print(np.random.choice(height, 100).mean( ))
```

163. 위에서 표본 100개를 뽑아서 평균값을 구해 a라는 비어있는 리스트에 담는 작업을 10000번 수행하시오!

```
import numpy as np

a=[]
height = np.random.randn(10000000) * 5 + 140
for i in range(10000):
    a.append(np.random.choice(height, 100).mean( ))
```

164. 위에서 구한 표본 평균값들 10000개의 평균값과 표준편차를 s_avg와 s_std변수에 각각 담으시오!

```
import numpy as np

a=[]
height = np.random.randn(10000000) * 5 + 140
for i in range(10000):
    a.append(np.random.choice(height, 100).mean( ))

s_avg = np.mean(a)
s_std = np.std(a)
print(s_avg) #139.9932309451772
print(s_std) #0.5021232291717785
```

165. 초등학생 키 데이터를 120부터 160까지 0.001 간격으로 생성해서 x라는 변수에 넣으

시오!

```
x=np.arange(120, 160, 0.001)
print(x)
```

166. 위에서 만든 x에 있는 키값들을 x축으로 두고 확률 밀도함수 그래프를 생성하는데 y축의 확률밀도함수값을 구할 때 문제 164번에서 구한 평균값, 표준편차를 이용하시오!

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

a=[]
height = np.random.randn(10000000) * 5 + 140
for i in range(10000):
        a.append(np.random.choice(height, 100).mean( ))

s_avg = np.mean(a)
s_std = np.std(a)
x=np.arange(138, 142, 0.001)
y=norm.pdf(x, s_avg, s_std)

plt.plot(x, y, color = 'blue')
plt.show( )
```

- ++ 오늘의 마지막문제

167. 총 클래스를 생성하는데 총 클래스로 아래와 같이 총(제품)을 생성하면 "총이 만들어졌습니다. 총알은 0발 장전되었습니다." 라는 메세지가 출력되게 총 클래스를 만드시오!

```
class Gun():
    def __init__(self):
        self.bullet = 0
        print('총이 만들어졌습니다. 총알은', self.bullet,'발 장전되었습니다.')
```

```
kms_gun1 = Gun()
```

168. 문자열 포멧을 이용해서 문제 167번의 메세지를 더 자연스럽게 출력되게 하시오!

```
class Gun():
    def __init__(self):
        self.bullet = 0
        print('총이 만들어졌습니다. 총알은 %d발 장전되었습니다.' %self.bullet)

kms_gun1 = Gun()
```

169. 충전한 금액보다 더 많은 금액을 소비하면 어떻게 되는지 테스트하시오!

```
class Card(): #부모 클래스
    def __init__(self):
        self.cash = 0
        print ('카드가 발급되었습니다.')

    def charge(self,num):
        self.cash += num
        print(num, '원이 충전되었습니다.')

    def consume(self,num):
        if self.cash >= num:
            self.cash -= num
            print(num, '원이 사용되었습니다.')
        else:
            print('잔액이 부족합니다.')

kms_card1 = Card()
kms_card1.charge(10000)
kms_card1.consume(20000)
```

```
>>> (결과)
카드가 발급되었습니다.
10000 원이 충전되었습니다.
잔액이 부족합니다.
```

170. 팀장님이 만든 card() 클래스를 상속받아 영화할인 카드를 생성하시오!

(영화 할인 카드 클래스: Movie_Card( ))

```
class Movie_Card( Card ): #부모클래스인 Card클래스를 상속받아 Movie_Card 클래스
를 만든다.
    pass

m_card1 = Movie_Card( )
m_card1.charge(100000)
m_card1.consume(8000)
```

171. 이번에는 제대로 영화관에서 사용하면 할인이 될 수 있도록 영화 클래스를 생성하시오!

```
class Movie_Card(Card):                 # 부모에게 __init__, charge, consume을 상속받았다.
    def consume(self, num, place):  #부모에게 받은 consume은 내가 만든 consume으로
        if place == '영화관':           # 덮어쓰기(overriding)가 된다.
            num = 0.8*num
            if self.cash >= num:       #잔액이 num보다 크다면 카드를 사용하고
                self.cash -= num
                print(place, '에서', num, '원이 사용되었습니다.')
            else:                           #아니면 잔액이 부족하다고 출력해라~
                print('잔액이 부족합니다.')

        else:                   #영화관에서 사용한게 아니라면 아래의 코드가 수행되게 해라
            if self.cash >= num:
                self.cash -= num
                print(num,'원이 사용되었습니다.')
            else:
                print('잔액이 부족합니다')

m_card1 = Movie_Card()
m_card1.charge(20000)
m_card1.consume(12000, '영화관')
m_card1.consume(2000, '편의점')
```

172. 171번의 영화 할인 카드에 할인 장소를 추가해서 주유소에서도 20% 할인 될 수 있도록 코드를 수정하시오!

```
class Movie_Card(Card):
    def consume(self, num, place):
        if place in ('영화관', '주유소') :
            num = 0.8*num
            if self.cash >= num:
                self.cash -= num
                print(place, '에서', num, '원이 사용되었습니다.')
            else:
                print('잔액이 부족합니다.')

        else:
            if self.cash >= num:
                self.cash -= num
                print(num,'원이 사용되었습니다.')
            else:
```

```
                print('잔액이 부족합니다')

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(12000, '영화관')
m_card1.consume(30000, '주유소')
m_card1.consume(2000, '편의점')
```

173. (점심시간 문제) 영화관과 주유소에서는 20% 할인되게 하고 스타벅스에서는 10% 할인되게 코드를 수정하시오!

```
class Movie_Card(Card):
    def consume(self, num, place):
        if place in ('영화관', '주유소') :
            num = 0.8*num
            if self.cash >= num:
                self.cash -= num
                print(place, '에서', num, '원이 사용되었습니다.')
            else:
                print('잔액이 부족합니다.')


        elif place == '스타벅스':
            num = 0.9*num
            if self.cash >= num:
                self.cash -= num
                print(place, '에서', num, '원이 사용되었습니다.')

        else:
            if self.cash >= num:
                self.cash -= num
                print(num,'원이 사용되었습니다.')
            else:
                print('잔액이 부족합니다')

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(12000, '영화관')
m_card1.consume(30000, '주유소')
m_card1.consume(6000, '스타벅스')
m_card1.consume(30000, '주유소')
m_card1.consume(2000, '편의점')
```

174. 목차 54번 예제에서 생성한 객체 emp_chulsu의 email 변수에 있는 내용을 출력해보시오.

```
print(emp_chulsu.email)    #chulsu.kim@gmail.com
```

175. 철수의 월급을 회사규정에 따라 인상시키시오!

emp_chulsu.apply_raise() #emp_chulsu 객체의 apply_raise() 메소드를 실행한다.
print(emp_chulsu.pay) # 5500000

176. 이번에는 새로운 사원 철수2로 emp_chulsu2 객체를 생성하시오!

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)

177. 철수2 사원이 월급을 인상시키기 위한 인스턴스 변수를 알아냈습니다.
자기는 월급을 10% 인상이 아니라 20% 인상 시키고 싶어서 아래와 같이
emp_chulsu2 객체의 멤버인 raise_amount라는 변수에 1.2를 할당하고 월급을 인상시키는 apply_raise 메소드를 실행을 했다.

```
emp_chulsu2.raise_amount = 1.2
print(emp_chulsu2.pay)   # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay)  # 6000000
```

178. 그래서 위와같이 악용이 될 수 없도록 막는 방법이 무엇입니까?

```
class  Employees: #옆에 괄호( )를 따로 안 쓴 이유는 밑의 __init__ 함수의 입력 매개변수가 여러개여서 입니다.
    raise_amount = 1.1  # 클래스 변수 (클래서 맴버) 클래스 변수에는 self가 없습니다.
    def __init__(self, first, last, pay): #emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
        self.first = first                  #위와 같이 객체(제품)를 생성할 때 입력값이 3개가 입력되면서
        self.last  = last                   #객체(제품)가 만들어 집니다.
        self.pay   = pay
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'
        self.raise_amount = 1.1  # 인스턴스 변수

    def  full_name(self): #사원의 전체이름을 출력하는 함수
        print  '{} {}'.format(self.first, self.last)

    def  apply_raise(self): # 월급을 인상하는 함수
        self.pay = int( self.pay * Employees.raise_amount)  # 클래스 변수를 사용
                                                  ↑
```

아까는 여기가 self였는데 지금은 클래스 이름인 Employees가 있다. 이자리에 인스턴스 변수가 아니라 클래스 변수를 사용했습니다.

```
emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)

emp_chulsu2.raise_amount = 1.2   # 인스턴스 변수만 변경할 수 있고 클래스 변수는
```

```
                                        # 변경할 수 없다.

print(emp_chulsu2.pay)   # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay)  # 5500000
```

설명 : 위와 같이 민감한 변수(멤버)들은 인스턴스 변수가 아니라 클래스 변수(멤버)를 사용해서 계산되게 코딩해야 합니다.

179. 판다스를 이용해서 emp3.csv에서 이름과 월급을 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']])
```

180. 이름이 SCOTT인 사원의 이름과 월급을 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['ename']=='SCOTT'])
```

181. input함수를 이용해서 이름을 물어보게 하고 이름 입력하면 해당 사원의 이름과 월급이 출력되게 하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['ename']=='SCOTT'])
#%%
import pandas as pd

name = input('이름을 입력하세요~')

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['ename']== name])
```

182. 이번에는 소문자로 이름을 입력해도 출력되게 하시오!

이름을 입력하세요~ scott

SCOTT 3000

답:

```
import pandas as pd

name = input('이름을 입력하세요~')

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['ename']== name.upper()])
```

또는

```
import pandas as pd

name = input('이름을 입력하세요~').upper()

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']][emp['ename']== name])
```

183. 이번에는 문제 182번의 코드를 실행하는데 없는 사원이름을 입력하시오!

```
이름을 입력하세요~ jack

결과:
Empty DataFrame
Columns: [ename, sal]
Index: []
```

184. 위의 코드에 예외처리를 해서 없는 사원이름을 입력하면 '입력하신 이름의 사원은 존재하지 않습니다.' 라는 메세지가 출력되게 하시오!

답 : 이것은 에러가 아니라서 예외처리가 되지 않습니다.

185. 숫자를 입력하면 해당 숫자의 제곱값이 출력되는 코드를 구현하시오!

```
숫자를 입력하세요~ 2
>>> 4
num = int(input('숫자를 입력하세요~'))
print(num**2)
```

186. try ~ except 예외처리를 이용해서 아래와 같이 문자를 입력하면 '잘못된 값을 입력하셨습니다'라는 메세지가 출력되게 하시오!

```
숫자를 입력하세요~ a
잘못된 값을 입력하셨습니다.

try:
    num = int(input('숫자를 입력하세요~'))
    print(num**2)

except:
    print('잘못된 값을 입력하셨습니다')
```

설명 : try와 except사이에 있는 코드에서 에러가 나야지만 except 이후의 문장을 실행합니다.

187. 아까 했던 나누기 프로그램을 수정해서 나누기가 성공하면 '성공적으로 나누기를 하였습니다.' 라는 메세지가 출력되게 하시오!

분자를 입력하세요~ 10
분모를 입력하세요~ 2

5
성공적으로 나누기를 하였습니다.

```
def my_devide():
    try:
        x = input('분자의 숫자를 입력하세요~')
        y = input('분모의 숫자를 입력하세요~')
        print(int(x)/int(y))
    except:
        print('당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.')
    else:
        print('성공적으로 나누기를 하였습니다')
print(my_devide())
```

188. 나누기하는 프로그램을 실행할 때 오류가 나던 오류가 나지 않던 무조건 아래의 메세지가 출력되게 하시오.

분자를 입력하세요~ 10 10
분모를 입력하세요~ 2 0

5 / 나누기를 할 수 없습니다.
이준혁이 만든 프로그램입니다. / 이준혁이 만든 프로그램입니다.

답:

```
def my_devide():
    try:
        x = input('분자의 숫자를 입력하세요~')
        y = input('분모의 숫자를 입력하세요~')
        print(int(x)/int(y))
    except:
        print('당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.')
    finally:
        print('이준혁이 만든 프로그램 입니다.')
print(my_devide())
```

189. 동전을 10번 던져서 앞면이 나올 확률을 출력하시오!

```
import random

coin = ['앞면', '뒷면']
cnt = 0
```

```
for k in range(1,10001):
    a = []              #실행문1
    for i in range(1,11):               #실행문2
        result=random.choice(coin)
        a.append(result)
# ['앞면', '뒷면', '뒷면', '뒷면', '앞면', '뒷면', '뒷면', '앞면', '앞면']
    if a.count('앞면') ==2: #실행문3
        cnt = cnt + 1

print(cnt/10000)
```

190. (오늘의 마지막 문제) 위의 코드를 함수로 생성하여 확률이 출력되게하시오!

```
coin_prob(2) #0.04314
coin_prob(4) #0.2065
              ↑
             앞면의 개수


def coin_prob(num1):
    import random

    coin = ['앞면', '뒷면']
    cnt = 0

    for k in range(1,10001):
        a = []
        for i in range(1,11):
            result = random.choice(coin)
            a.append(result)

        if a.count('앞면') == num1:
            cnt = cnt +1
    return cnt/10000

print(coin_prob(2))
print(coin_prob(4))
```

191. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 1번부터 출력되게하는 코드를 작성하시오!

```
숫자를 입력하세요~ 5


1
2
3
4
5
```

답:

```
a = int(input('숫자를 입력하세요'))
for i in range(1,a+1):
    print(i)
```

192. 문제 191번의 코드에 예외처리를 해서 숫자를 물어볼 때 문자를 입력하면 잘못된 값을 입력하셨습니다. 라고 메세지가 출력되게 하시오.

```
try:
    a = int(input('숫자를 입력하세요'))
    for i in range(1,a+1):
        print(i)
except:
    print('잘못된 값을 입력하셨습니다')
```

설명 : 위의 코드의 경우에는 숫자를 입력할 대 알파벳 a를 넣으면 예외처리가 되어서 잘못됐 값을 입력하셨습니다. 라는 말만 나오고 에러에 대한 정확한 원인 파악은 하기가 어렵습니다.

잘못된 값을 입력하셨습니다 말고도 정확한 에러에 대한 원인을 파악하고 싶다면 아래와 같이 작성하면 됩니다.

```
try:
    a = int(input('숫자를 입력하세요'))
    for i in range(1,a+1):
        print(i)
except Exception as e:
    print('잘못된 값을 입력하셨습니다')
    print(e) #에러가 난 이유를 출력해줍니다.
```

```
숫자를 입력하세요  a
잘못된 값을 입력하셨습니다
invalid literal for int() with base 10: 'a'
```

193. 판다스를 이용해서 emp3.csv의 데이터를 로드하는데 이름을 물어보게 하고 이름을 입력하면 해당 사원의 이름과 월급이 출력되게 하시오!

이름을 입력하세요~ scott
SCOTT 3000

```
import pandas as pd

name = input('이름을 입력하세요~')

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
```

```
print(emp[['ename', 'sal']][emp['ename'] == name.upper()])
```

194. 위의 결과에서 월급만 출력하시오!

```
import pandas as pd

name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['sal']][emp['ename'] == name.upper()])
   sal
11  3000
-----------------------
import pandas as pd

name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp['sal'][emp['ename'] == name.upper()]
print(type(result))

<class 'pandas.core.series.Series'>
-----------------
값으로만 나오게 하고싶다?
import pandas as pd

name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp['sal'][emp['ename'] == name.upper()].values[0]
print(type(result))

---
<class 'numpy.int64'>
-----
import pandas as pd

name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp['sal'][emp['ename'] == name.upper()].values[0]
print(result)
>>>
3000
```

설명 : emp데이터 프레임에서 어떤 특정값을 딱 하나만 출력하려면 위와 같이 작성해 줘야 합니다.

195. 위의 코드에 사용자 정의 예외처리를 해서 월급이 고소득자는 해당사원의 월급을 볼 수 없습니다. 라는 메세지가 출력되게 하시오!
(월급이 3000이상인 사원들을 고소득자로 보고 작성하시오)

```
import pandas as pd
```

```
name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp['sal'][emp['ename'] == name.upper()].values[0]
if result >= 3000:
    raise Exception('해당 사원의 월급은 볼 수 없습니다')
print(emp[['ename', 'sal']][emp['ename']==name.upper()])
```

196. 문제 195번의 코드를 수정해서 이름을 물어보게하고 이름과 직업을 출력하는 코드로 작성하는데 직업이 SALESMAN이면 '해당 사원의 정보는 볼 수 없습니다.' 라는 메세지가 출력되면서 프로그램이 종료되게 하시오!

```
import pandas as pd

name = input('이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp['job'][emp['ename'] == name.upper()].values[0]
if result =='SALESMAN':
    raise Exception('해당 사원의 월급은 볼 수 없습니다')
print(emp[['ename', 'job']][emp['ename']==name.upper()])
```

—

```
사원 이름을 입력하세요~aaa
Empty DataFrame
Columns: [ename, sal]
Index: []
```

없는 사원의이름을 넣으면 이렇게 나온다.

197. 아래처럼 사원 이름을 물어보게하고 사원이름을 입력하면 해당 사원의 이름과 월급이 출력되게하시오!

```
import pandas as pd
name = input('사원 이름을 입력하세요~')
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal''[emp['ename']==name.upper()])
```

198. 없는 사원 이름을 입력하고 해당사원은 없습니다~라는 문구가 나오게 하시오.

```
import pandas as pd

try:

    name = input('사원 이름을 입력하세요~')
    emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
    result = emp['ename'][emp['ename']==name.upper()].values[0]
    print(emp[['ename', 'sal']][emp['ename']==result])

except LookupError:
    print('해당 사원은 없습니다')
```

설명 : result = emp['ename'][emp['ename']==name.upper()].values[0] 이 코드에서 values[0]을 사용하면 Series(컬럼)가 아니라 값으로 출력이 되어서 result에 담기게 됩니다. 없는 사원이름을 입력하면 result에 값이 입력되지 않게 되므로 LookupError 예외처리가 되어서 '해당사원은 없습니다. ' 라는 메세지가 출력되는 겁니다.

```
>>>
사원 이름을 입력하세요~dd
해당 사원은 없습니다
```

199. (점심시간 문제) 직업을 물어보게하고 직업을 입력하면 해당 사원의 이름과 직업과 월급이 출력되게하는 코드를 작성하는데 없는 직업을 입력하면 '해당 직업은 사원 테이블에 없습니다.' 라는 메세지가 출력되게하시오!

```
직업을 입력하세요~ salesman
직업을 입력하세요 ~ aaaa

해당 직업은 사원 테이블에 없습니다.
```

```
import pandas as pd

try:

    jname = input('사원의 직업을 입력하세요~')
    emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
    result = emp['job'][emp['job']==jname.upper()].values[0]
    print(emp[['ename', 'sal', 'job']][emp['job']==result])

except LookupError:
    print('해당 직업은 없습니다')
```

200. 딕셔너리 자료형을 만들고 61번 인덱스 예제와 같이 type을 확인하시오!

```
b = {'사과':'apple', '배':'pear'}
print(type(b)) #<class 'dict'>
```

설명 : 리스트 --> 대괄호 [ ]
튜플 --> 소괄호( )
딕셔너리 --> 중괄호 { }

201. 아래와 같이 두개의 숫자를 각각 물어보게 하고 아래의 메세지가 출력되게하시오!

첫번째 숫자를 입력하세요 ~ 1113

두번째 숫자를 입력하세요~ 23

1113을 23으로 나누면 9가 나머지로 남습니다.

답:

```
a = int(input('첫번째 숫자를 입력하세요~'))
b = int(input('두번째 숫자를 입력하세요~'))

print(a,'을',b,'로 나누면',a%b,'가 나머지로 남습니다')
```

또는

```
a = int(input('첫번째 숫자를 입력하세요~'))
b = int(input('두번째 숫자를 입력하세요~'))
c = a%b

print('%d을 %d로 나누면 %d가 나머지로 남습니다' %(a,b,c))
```

202. 아래와 같이 실행되게 코드를 수행하시오!

첫번째 숫자를 입력하시오~ 1113

두번째 숫자를 입력하시오~ 23

1113을 23으로 나눈 몫은 48이고 나머지는 9입니다.

첫번째 숫자를 입력하시오~ 1113

두번째 숫자를 입력하시오~ 0

0으로는 나눌 수 없습니다.

답:

```
try:

    a = int(input('첫번째 숫자를 입력하세요~'))
    b = int(input('두번째 숫자를 입력하세요~'))
    result1, result2 = divmod(a,b)
    print('%d을 %d으로 나눈 몫은 %d이고 나머지는 %d입니다' %(a,b,result1, result2))
except ZeroDivisionError:
    print('0으로는 나눌 수 없습니다')
```

203. dept3.csv를 판다스로 로드해서 dept데이터 프레임 전체를 출력하시오!

```
import pandas as pd

dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
print(dept)
```

204. 부서위치가 DALLAS의 부서번호와 부서명(dname)을 출력하시오!

```
import pandas as pd

dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
print(dept[['deptno', 'dname']][dept['loc']=='DALLAS'])
```

205. DALLAS에서 근무하는 사원들의 이름과 부서위치를 출력하시오!

SQL >

```
select e.ename, d.loc
      from emp e, dept d
      where d.loc = 'DALLAS';
```

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
result = pd.merge(emp, dept, on = 'deptno')

print(result[['ename', 'loc']][result['loc']=='DALLAS'])
```

206. 월급이 3000이상인 사원들의 이름과 월급과 부서위치를 출력하시오

SQL >

```
select e.ename, e.sal, d.loc
      from emp e, dept d
      where sal >= 3000;
```

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
result = pd.merge(emp, dept, on = 'deptno')

print(result[['ename', 'sal', 'loc']][result['sal']>=3000])
```

207. 부서번호가 10번, 20번인 사원들의 이름과 부서위치와 부서번호를 출력하시오!

SQL>

```
select e.ename, d.loc, d.deptno
      from emp e, dept d
      where deptno in (10,20);
```

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
result = pd.merge(emp, dept, on = 'deptno')

print(result[['ename', 'loc', 'deptno']][emp['deptno'].isin([10,20])])
```

208. 월급이 1000에서 3000사이인 사원들의 이름과 월급과 부서위치를 출력하시오!

SQL>

```
select e.ename, e.sal, d.loc
      from emp e, dept d
```

where sal between 10000 and 3000;

Pandas>
```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")
result = pd.merge(emp, dept, on = 'deptno')

print(result[['ename', 'sal', 'loc']][emp['sal'].between(1000,3000)])
```

209. 아래의 SQL을 Pandas로 구현하시오. (아우터조인)

SQL>
```
select e.ename, d.loc
    from emp e, dept d
    where e.deptno(+) = d.deptno;
```

Pandas>
```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, on = 'deptno', how = 'right')
print(result[['ename', 'loc'] ])
```

210. 아래의 SQL을 Pandas로 구현하시오!

SQL>
```
select e.ename, d.loc
    from emp e full outer join dept d
    on( e.deptno = d.deptno);
```

Pandas>
```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, on = 'deptno', how = 'outer')
print(result[['ename', 'loc']])
```

211. 아래의 서브쿼리를 Pandas로 구현하시오.

SQL>
```
select ename, sal
        from emp
        where job = (select job
                        from emp
                        whre ename ='SCOTT');
```

답 :

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

sjob = emp['job'][emp['ename']=='SCOTT'].values[0]
print(emp[['ename', 'sal']][emp['job']== sjob])
```

212. 아래의 서브쿼리의 결과를 Pandas로 수행하시오!

```
SQL > select ename, sal
            from emp
            where job = ( select job
                          from emp
                          where ename ='SCOTT')
            and ename !='SCOTT';
```

답:

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

sjob = emp['job'][emp['ename']=='SCOTT'].values[0]
print(emp[['ename', 'sal']][(emp['job']== sjob) & (emp['ename'] != 'SCOTT')])
```

설명 : 판다스에서 and는 &이고 or는 | 입니다.
& 와 |를 쓸때는 괄호로 묶어줘야 합니다.
~은 not이고 != 은 같지 않다는 연산자

213. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select max(sal)
            from emp
            where deptno = 20;
```

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['sal'][emp['deptno']==20].max())
```

214. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select min(sal)
            from emp
            where job = 'SALESMAN';
```

답:

Pandas>

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['sal'][emp['job']=='SALESMAN'].min())
```

215. emp12.csv를 판다스 데이터 프레임으로 만들어서 출력하시오!

```
import pandas as pd

emp12 = pd.read_csv("c:₩₩data₩₩emp122.csv")
print(emp12)
```

216. 우리반에서 최소 나이를 출력하시오!

```
import pandas as pd

emp12 = pd.read_csv("c:₩₩data₩₩emp122.csv")
print(emp12['AGE'].min())
```

217. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select job, max(sal)
            from emp
            group by job;
```

답:

Pandas>

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp.groupby('job')['sal'].max()
print(result)
```

```
job
ANALYST       3000
CLERK         1300
MANAGER       2975
PRESIDENT     5000
SALESMAN      1600
Name: sal, dtype: int64
<class 'pandas.core.series.Series'>
----------------------
```

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp.groupby('job')['sal'].max().reset_index()
print(type(result))
```

해야

<class 'pandas.core.frame.DataFrame'>

타입이 바뀐다.

※ 설명 : reset_index() 키워드는 Series로 출력하는게 아니라 DataFrame(테이블)으로 출력하는 키워드

입니다.

```
         job   sal
0    ANALYST  3000
1      CLERK  1300
2    MANAGER  2975
3  PRESIDENT  5000
4   SALESMAN  1600
<class 'pandas.core.frame.DataFrame'>
```

218. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select deptno, sum(sal)
            from emp
            group by deptno;
```

답:

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()
print(result)
```

219. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select deptno, sum(sal)
            from emp
            where deptno !=20
            group byy deptno;
```

답:

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()
print(result[['deptno', 'sal']][result['deptno']!=20])
```

---

emp[컬럼][조건]

result[컬럼][조건]

220. (오늘의 마지막 문제) 어제 마지막 문제로 만든 함수를 이용해서 아래와 같이 출력되게 하시오!

```
def coin_prob(num1):
    import random

    coin = ['앞면', '뒷면']
    cnt = 0

    for k in range(1,10001):
        a = []
        for i in range(1,11):
            result = random.choice(coin)
            a.append(result)

        if a.count('앞면') == num1:
            cnt = cnt +1
    return cnt/10000

print(coin_prob(2))
print(coin_prob(4))
```

221. 부서번호, 부서번호별 평균월급을 판다스로 출력하시오!

```
select deptno, avg(sal)
       from emp
       group by deptno;
```

답:

```
Pandas
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('deptno')['sal'].mean().reset_index()
print(result)
```

---

reset_index( )가 한 세트이다.

설명 : groupby부터 시작해서 reset_index( )까지가 한 세트이므로 항상 같이 사용하면 됩니다.

결과:

	deptno	sal
0	10	2916.666667
1	20	2175.000000
2	30	1566.666667

222. 문제 221번의 결과에서 평균월급을 출력할 때 정수부분만 출력하고 싶다면? (소수점 이전만 출력)

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('deptno')['sal'].mean().reset_index().astype(int)
print(result)
```

결과:

	deptno	sal
0	10	2916
1	20	2175
2	30	1566

설명: astype(int)는 출력되는 데이터에서 정수부분만 취해라 라는 뜻입니다.
정수형으로 변환해라~ 라는 뜻입니다.
astype(float)는 실수로 출력합니다.

223. 직업과 직업별 토탈월급을 출력하시오! (판다스로)

```
SQL > select job, sum(sal)
        from emp
        group by job;
```

답:

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('job')['sal'].sum().reset_index()
print(result)
```

224. 부서위치. 부서위치별 토탈월급을 출력하시오!

SQL > select d.loc, sum(e.sal)

```
            from emp e, dept d
            where e.deptno = d.deptno
            group by d.loc;
```

답:

```
Pandas>
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, on = 'deptno')
result2 = result.groupby('loc')['sal'].sum().reset_index()
print(result2)
```

225. 아래의 SQL을 판다스로 구현하시오!

```
SQL > select d.loc, sum(e.sal)
            from emp e, dept d
            where e.deptno(+) = d.deptno
            group by d.loc;
```

답:

```
Pandas>
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, how = 'right', on = 'deptno')
result2 = result.groupby('loc')['sal'].sum().reset_index()
print(result2)
```

226. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select deptno, count(*)
            from emp
            group by deptno;
```

답:

```
Pandas>
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('deptno')['empno'].count().reset_index()
print(result)
```

pandas에서는 *가 없기 때문에 뒤에 null값이 없는 값으로 묶어주기만 하면 된다.

227. emp122.csv를 내려받아 판다스 데이터 프레임으로 만들고 출력하시오!

```
import pandas as pd

emp122=pd.read_csv("c:₩₩data₩₩emp122.csv")
```

```
print(emp122)
```

228. 통신사, 통신사별 인원수를 출력하시오!

```
import pandas as pd
emp122 = pd.read_csv("c:₩₩data₩₩emp122.csv")
result = emp122.groupby('TELECOM')['INDEX'].count().reset_index()
print(result)
```

결과:

```
  TELECOM  INDEX
0      kt     15
1      lg      4
2      sk     11
```

229. 우리반 테이블에서 통신사가 kt이고 나이가 30살 이상인 학생들의이름과 나이와 통신사를 출력하시오!

```
import pandas as pd
emp122 = pd.read_csv("c:₩₩data₩₩emp122.csv")
print(emp122[['ENAME', 'AGE', 'TELECOM']][(emp122['TELECOM'] == 'kt') &
(emp122['AGE']>=30)])
```

설명 : and는 판다스에서 &이고 or는 판다스에서 | 입니다.
그리고 & 와 | 를 사용할 때는 양쪽 조건에 소괄호를 둘러줘야 합니다.

230. 판다스를 이용하지 말고 이름과 월급*12.3를 출력하시오.

```
import csv
file = open("c:₩₩data₩₩emp2.csv") #os에 있는 emp2.csv를 읽어서 file이라는 변수에 넣는다
emp_csv = csv.reader(file) #file변수에 있는 csv파일을 읽어서 emp_csv 변수에 넣는다
print(emp_csv) #이상태에서 그냥 프린트하면 메모리 주소가 나옵니다.

for emp_list in emp_csv: #csv파일의 내용을 한행씩 리스트에 담아서
    print(emp_list[1], int(emp_list[5])*12.3)   #출력합니다
```

설명 : csv파일의 내용을 읽어서 출력할 때 기본적으로 월급도 문자형으로 출력되므로 산술연산을 하려면 int()함수를 이용해서 숫자형으로 변환해주어야 합니다.

231. 문제 230번의 결과를다시 출력하는데 소수점 첫번째 자리에서 반올림되게 하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
print(emp_csv)

for emp_list in emp_csv:
    print(emp_list[1], round(int(emp_list[5])*12.3))
```

232. 직업이 SALESMAN인 사원들의 이름과 직업을 출력하는데 판다스 이용하지 말고 emp2.csv파일로 읽어서 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
print(emp_csv)

for emp_list in emp_csv:
    if emp_list[2] == 'SALESMAN':
        print(emp_list[1], emp_list[2])
```

233. (12/7 점심시간 문제) 부서번호가 20번인 사원들의 이름과 월급과 부서번호를 출력하시오~

판다스를 이용한 방법:

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

print(emp[['ename', 'sal', 'deptno']][emp['deptno']==20])
```

판다스를 이용하지 않은 방법:

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
print(emp_csv)

for emp_list in emp_csv:
    if emp_list[7] == '20':
        print(emp_list[1], emp_list[5], emp_list[7])
```

234. 판다스를 이용해서 emp3.csv를 읽어다가 이름과 월급을 출력하는데 월급을 출력할 때 소수점 이하는 버리고 정수 부분만 출력되게 하시오!

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[['ename', 'sal']])
```

설명 : int( )함수를 따로 안써도 정수형으로 출력되고 있습니다.
판다스를 이용하지 않았을때와는 다르게 숫자는 바로 숫자형으로 출력해주고 있습니다.

235. 판다스를 이용해서 이름과 월급을 출력하는데 월급을 출력할 때 실수형으로 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

emp['sal'] = emp['sal'].apply(float) #emp 데이터프레임의 sal컬럼의 데이터를 float(실
```

수형)로 변환해서

#emp 데이터프레임에 sal컬럼의 데이터로 변경해라~

```
print(emp[['ename','sal']])
```

설명 : emp['sal'].apply는 emp 데이터 프레임에 sal 시리즈에 apply함수를 적용해서 데이터 유형을 변경할 수 있습니다.
emp['sal'].apply(float) 이렇게 하면 데이터 유형을 실수형으로 변경하는 것입니다.

236. 아래의 리스트에서 숫자가 300이상이면 출력하고 300미만이면 출력되지 않게 하시오!

```
b = [100, 352, 254, 456, 123, 234, 567, 903]

def get_over(num):
    if num >=300:
        return num
    else:
        return

result = filter(get_over, b)
print(list(result))  #[352, 456, 567, 903]
```

237. 우리반 데이터에서 나이가 30살 이상인 나이만 따로 결과로 리스트를 출력하시오!
[31, 30.....]

```
import pandas as pd

emp122 = pd.read_csv("c:₩₩data₩₩emp122.csv")
a = []
for i in emp122['AGE']:
    if i >=30:
        a.append(i)

print(a)
```

238. 사원 테이블에서 최대월급을 출력하시오! (emp3.csv)

1. **판다스를 이용했을 때**

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp['sal'].max())
```

2. **판다스를 이용하지 않았을 때**

```
import csv

file = open("c:₩₩data₩₩emp2.csv")
```

```
emp = csv.reader(file)
a=[]
for i in emp:
    a.append(int(i[5]))

print(max(a))
```

239. 우리반 데이터(emp122.csv)에서 최소나이를 출력하시오!

1. 판다스 이용했을때

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp122.csv")

print(emp['AGE'].min())
```

2. 판다스 이용하지 않았을 때

emp122.csv를 북사해서 emp1222.csv로 붙여넣는다.

```
import csv

file = open("c:₩₩data₩₩emp1222.csv")
emp12 = csv.reader(file)
a = []
for i in emp12:
    a.append(int(i[2]))
print(min(a))
```

240. 커미션이 결측치(NaN)인 사원들의 이름과 커미션을 출력하시오.

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[ ['ename', 'comm'] ][emp['comm'].isnull()])
```

241. 커미션이 결측치(NaN)이 아닌 사원들의 이름과 커미션을 출력하시오!

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[ ['ename', 'comm'] ][~emp['comm'].isnull()])
```

*데이터를 받았으면 데이터 분석을 하기 전에 데이터 전처리를 해야하는데 전처리 중에서 결측치 확인하는 단계가 있습니다.

242. emp3.csv에 결측치가 있는지 확인하는 방법?

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp[ ['ename', 'comm'] ][~emp['comm'].isnull()])
#%%
import pandas as pd
```

```
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp.isnull())
```

243. emp3.csv에 결측치가 몇개인지 확인하는 방법?

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
print(emp.isnull().sum())
```

결과:

```
index       0
empno       0
ename       0
job         0
mgr         1
hiredate    0
sal         0
comm       10
deptno      0
```

설명 : 결측치가 있으 데이터 분석하기가 어렵고 머신러닝을 이용한 데이터 분석인 경우 좋은(정확도가 높은) 머신러닝 모델이 나오기가 어렵습니다. 결측치를 처리를 해줘야 합니다.

244. 타이타닉 데이터에 결측치가 어느 컬럼에 많은지 확인하시오!

```
import pandas as pd

train = pd.read_csv("c:₩₩data₩₩train.csv")
print(train.isnull().sum())
```

결과:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
```

245. 판다스를 이용해서 이름과 부서위치를 출력하시오!

```
import pandas as pd
```

```
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, how = 'inner', on = 'deptno')
print(result[['ename', 'loc']])
```

246. emp 데이터 프레임 loc컬럼을 추가하고 해당 사원의 부서위치로 값을 갱신하시오!

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
dept = pd.read_csv("c:₩₩data₩₩dept3.csv")

result = pd.merge(emp, dept, how = 'inner', on = 'deptno')
emp['loc'] = result['loc'] #emp데이터프레임에 loc컬럼을 추가하면서 result의 loc로 값을 갱신합니다.

print(emp)
```

설명: 파생변수를 왜 추가 하냐면?

예제 : emp테이블에서 퇴사할 것 같은 사원이 누구인지 예측하시오!

--> 머신러닝을 이용해서 예측을 하면 됩니다.

머신러닝이 예측을 잘하려면 좋은 데이터를 주고 학습시켜야 합니다.

자기의 월급이 자기가 속한 직업의 평균월급보다 더 작은 월급을 받는 사원이면 퇴사할 가능성이 높다.

직업별 평균월급이 emp데이터 프레임에 추가 되어 있으면 머신러닝이 예측하기 좋은 데이터가 추가가 된것 입니다.

247. 직업, 직업별 평균월급을 판다스로 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('job')['sal'].mean().reset_index()

result['sal'] = result['sal'].astype(int) #result 데이터프레임에 sal을 정수형으로 변환해서
                                          #result데이터프레임의 sal에 반영하겠다.
print(result)
```

결과:

```
        job          sal
0   ANALYST   3000.000000
1     CLERK   1037.500000
2   MANAGER   2758.333333
3  PRESIDENT   5000.000000
```

4 SALESMAN 1400.000000

248. emp와 result를 서로 조인해서 조인된 전체 데이터 프레임을 출력하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('job')['sal'].mean().reset_index()

result['sal'] = result['sal'].astype(int)
```

3.

.....
.....
.....

설명: emp에도 sal이 있고 result에도 sal이 있어서 emp의 sal은 컬럼명이 sal_x로 변경되었고 result의 sal은 sal_y로 변경되었습니다. sal_y는 해당 직업의 평균월급입니다.

249. emp 데이터 프레임에 컬럼을 하나 추가하는데 job_avgsal로 추가하고 문제 248번에서 구한 직업별평균월급인 result2['sal_y']의 값으로 값을 갱신하시오!

```
import pandas as pd
emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result = emp.groupby('job')['sal'].mean().reset_index()

result['sal'] = result['sal'].astype(int)

result2 = pd.merge(emp, result, how = 'inner', on = 'job')
emp['job_avgsal'] = result2['sal_y']
print(emp)
```

설명 : 현업에서 머신러닝 데이터 분석가들이 하는 일중 상당수가 바로 이런 파생변수를 추가하는 직업입니다. 좋은 파생변수를 추가해야 머신러닝이 예측을 잘 할 수 있습니다.

Ex : 게임회사에서의 응용방법: 그 게임을 이탈할 것 같은 유져를 머신러닝으로 찾아서 형평성에 어긋나지않도록하면서 그 유져가 인식하지 못하도록 조용히 혜택을 줍니다.

250. emp 데이터 프레임에 해당 사원이 근무하는 부서번호의 평균 월급을 sal_avg라는 이름으로 파생변수를 생성하시오.

(내일 답 알려주실 예정)

251. 판다스로 emp데이터 프레임을 출력하는데 월급이 높은 사원부터 출력하시오!

```
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")
result = emp.sort_values('sal', ascending = False)
print(result)
```

설명 : 판다스를 사용할 때 데이터를 정렬하려면 위와 같이 sort_values함수를 이용하

면 됩니다.

ascending = True : 낮은값에서 높은 값 순으로 정렬하겠다.

ascending = False : 높은값에서 낮은값 순으로 정렬하겠다.

252. (오늘의 마지막 문제) 아래의 SQL을 판다스로 구현하시오!

```
SQL > select job, sum(sal)
            from emp
            group by job
            having sum(sal) >= 6000
            order by sum(sal) desc;
```

답:

```
Pandas>
import pandas as pd

emp = pd.read_csv("c:₩₩data₩₩emp3.csv")

result1 = emp.groupby('job')['sal'].sum().reset_index()
result1['sal']=result1['sal'].astype(int)

result2 = result1[['sal', 'job']][result1['sal']>=6000]
result3 = result2.sort_values('sal', ascending = False)
print(result3)
```

253. 아래의 txt변수에서 w를 출력하시오!

```
txt ='A tale that was not right'
print(txt[12])  #w
```

254. 위의 txt변수에서 맨 끝의 철자인 t를 출력하시오!

```
txt ='A tale that was not right'
print(txt[-1])
```

255. emp3.csv에서 이름만 출력하시오

1. 판다스를 이용하지 않았을 때

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1])
KING
BLAKE
CLARK
JONES
MARTIN
ALLEN
TURNER
JAMES
WARD
FORD
```

SMITH
SCOTT
ADAMS
MILLER

256. 255번 문제에서 출력된 이름의 데이터 유형이 무엇인지 확인하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(type(emp_list))
```

```
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
```

257. 문제255번을 다시 수행해서 emp2.csv 에서 이름을 출력하는데 이름의 첫번째 철자만 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1][0]) #a[0]
```

```
K
B
C
J
M
A
T
J
W
F
S
S
A
M
```

258. 문제257번의 결과를 다시 출력하는데 첫번째 철자가 소문자로 출력되게 하시오.

```
import csv
```

```
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1][0].lower()) #a[0]
```

```
k
b
c
j
m
a
t
j
w
f
s
s
a
m
```

259. 아래의 SQL을 파이썬으로 구현하시오!
(판다스 이용하지 말고 그냥 emp2.csv를 읽어서 구현)

```
SQL> select ename, substr(ename, 1, 3)
            from emp;
```

파이썬>

```
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1][0:3])
```

260. 다음 txt문자열에서 짝수번째 철자들만 출력하시오!

```
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
result3 = txt[1::2]
print(result3)
```

261. 이름을 출력하는데 이름의 철자를 거꾸로 출력하시오!
(emp2.csv를 읽어서 하세요~ 파이썬으로만 하세요)

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1][::-1])
```

262. 아래의 SQL울 파이썬으로 구현하시오! (emp2.csv를 이용하세요)
SQL > select ename||job

from emp;

답:

```
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1] + emp_list[2])
```

결과:

```
KINGPRESIDENT
BLAKEMANAGER
CLARKMANAGER
JONESMANAGER
MARTINSALESMAN
ALLENSALESMAN
TURNERSALESMAN
JAMESCLERK
WARDSALESMAN
FORDANALYST
SMITHCLERK
SCOTTANALYST
ADAMSCLERK
MILLERCLERK
```

263. 아래와 같이 결과가 출력되게 하시오!

SQL> select ename||'의 직업은' ||job|| '입니다.'
from emp;

답:

```
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1]+'의 직업은 '+ emp_list[2] + ' 입니다')
```

264. for loop문을 이용해서 숫자 1부터 5까지 출력하시오!

```
1
2
3
4
5
```

답:

```
for i in range(1,6)
      print(i)
```

265. 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자까지 숫자가 출력되게하시오!

숫자를 입력하세요~5

1
2
3
4
5

답:

```
a = int(input('숫자를 입력하세요~'))
for i in range(1,a+1):
    print(i)
```

266. 위의 코드를 수정해서 숫자를 물어보게하고 아래와 같이 ★이 출력되게하시오!

숫자를 입력하시오~ 5

★
★★
★★★
★★★★
★★★★★

답:

```
a = int(input('숫자를 입력하세요~'))
for i in range(1,a+1):
    print(i*'★')
```

267. (알고리즘 22번문제) 아래와 같이 사각형이 출력되게 하시오!

가로의 숫자를 입력하세요~ 3
세로의 숫저를 입력하세요~ 5

```
a = int(input('숫자를 입력하세요~'))
for i in range(1,a+1):
    print(i*'★')
```

268. emp2.csv에서 이름을 출력하는데 이름에 S를 포함하고 있는 사원들의 이름을 출력하시오!

```
SQL> select ename
            from emp
            where ename like '%S%';
```

답:

파이썬>

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
```

```
    if 'S' in emp_list[1]:
        print(emp_list[1])
```

결과:
JONES
JAMES
SMITH
SCOTT
ADAMS

설명: 위의 코드는 금융감독원에서 금융사에 요청하는 것중에 하나가 보험회사의 경우 보험 상담원들이 상담할 때 적절한 보험용어를 써서 고객을 응대를 해야해서 고객과의 통화를 다 녹음을 하고 그 녹음된 내용을 text로 변환해서 그 text안에 단어들을 일일히 확인하는 작업을 한다.
(예 : 30일이라고 해야하는데 한달이라고 했다던가 할 때) 이럴 때 응용될 수 있는 코드입니다.

269. 문제 268번 코드를 수정해서 출력하는데 이름에 S자가 포함된 사원의 이름이 몇명인지 출력하시오!
결과 : 5

답:
```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
cnt = 0
for emp_list in emp_csv:
    if 'S' in emp_list[1]:
        cnt = cnt+1
print(cnt)
```

270. (점심시간 문제) 겨울왕국 스크립트에서 elsa라는 문자열(단어)이 몇번 나오는가?
오전에 배웠던 내용으로만 elsa가 몇건 들어있는지 카운트
틀린답(오전에 배운 내용으로만 했을때) :
```
winter = open('c:₩₩data₩₩winter.txt')
winter2 = winter.read().split(' ')
cnt = 0

for i in winter2:
    if 'elsa' in i.lower():
        cnt = cnt+1
        print(i.lower(),cnt)
```

elsa 304
confidence. anna elsa? elsa love. elsa 305

설명: 위의 결과를 보면 304 다음에 elsa 가 3개가 있으므로 305가 되면 안되는데 305가 된 이후는 confidence. anna elsa? elsa love. elsa 를 하나의 문장으로 보았기 때문입니다. 어절별로 분리한걸로 코드를 구현했지만 실제로는 어절별로 분리가 되지 않았습니다.
그래서 위와같은 실수를 하지 않으려면 해결하는 방법은?
먼저 스크립트를 행단위로 먼저 분리를 하고나서 어절별로 분리를 해야 합니다.

답1:

```
winter = open("c:₩₩data₩₩winter.txt")
winter2 = winter.read().split("₩n") #행단위로 분리하는 코드

cnt = 0
for i in winter2:
    for j in i.split(' '):
        if 'elsa' in j.lower():
            cnt +=1
            print(j,cnt)
```

또는

답2:

```
winter = open("c:₩₩data₩₩winter.txt")
winter2 = winter.read().split() #아무것도 안쓰면 어절별로 나뉜다.
cnt = 0
for i in winter2: #어절별로 분리한 것들을 하나씩 가져오면서
    if 'elsa' in i.lower(): #소문자로 변경하고 elsa가 포함되어져 있는지 확인
        cnt +=1 #하면서 cnt를 누적히키며 증가시킨다.
print(cnt)
```

271. 파이썬으로 emp2.csv를 읽어 이름, 이름의 길이를 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1], len(emp_list[1]))
```

272. 스티븐 잡스 연설문인 jobs.txt를 한행 한행씩 출력하시오!

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split('₩n') #한 행씩 분리하는 작업
for i in stev2:                  #한 행씩 분리된 스크립트를 읽어서
    print(i)                     #한 행씩 출력한다.
```

설명: 위의 코드를 실행했는데 cp949 에러가 나면 encoding에 UTF8도 써보고 ANSI도 써보고 CP949도 써보세요~

273. 위에서 한행씩 출력하고 있는 스크립트를 철자 하나씩 출력하시오.

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split('₩n')
for i in stev2:
    for k in i: #스크립트 한행을 읽어서 철자를 하나씩 불러오는 코드
        print(k) #그 철자를 출력한다.
```

274. 문제 273번에서 출력된 철자가 알파벳이면 cnt를 증가시켜서 스티븐 잡스 연설문에 알파벳이 몇개가 있는지 출력하시오!

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split('₩n')
cnt = 0
for i in stev2:
    for k in i:
        if k.isalpha()==True:
            cnt = cnt+1
print(cnt)
```

설명: 스티븐 잡스 연설문에서 숫자나 공백문자나 마침표와 같은 구두점을 뺀 알파벳 철자만 카운트 했습니다.
데이터 분석 사례중에 유명한 사례중 하나가 아프리카 케냐의 은행에서 고객들에게 대출을 해 줄때 그 사람의 신용을 확인해서 대출을 해 주는데 신용을 확인할 때 sns의 그 사람에 대한 글을 분석해서 긍정적인 평가가 많은지 부정적인 평가가 많은지를 파악해서 대출심사에 반영을 하니까 훨씬 은행의 대출금 환수가 원활해졌던 사례에서 사용되고 있습니다.

275. 스티븐잡스 연설문에는 숫자가 몇개가 있는지 출력하시오.

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split('₩n')
cnt = 0
for i in stev2:
    for k in i:
        if k.isdigit()==True:
            cnt = cnt+1
print(cnt)
```

276. 스티븐 잡스 연설문에는 긍정단어가 몇개가 있는지 확인하세요~ 코드를 답글로 올리세요~

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split() #스티븐잡스 연설문을 어절별로 분리하여
                            #stev2에 입력햇는데 stev2도 리스트이고 stev2리스트에는
                            #스티븐 잡스 연설문이 어절별로 분리되어서 저장되어 있습
니다.
positive = open("c:₩₩data₩₩positive-words.txt")
pos = positive.read().split('₩n') #positive에서 행별로 분리해서 pos에 담는데 pos는
리스티입니다.
```

```
                                    # pos 리스트에는 긍정단어들이 들어있습니다.
cnt = 0
for i in stev2:          #stev3리스트에서 하나씩 빼냅니다.
    if i.lower() in pos:          #i의 단어가 pos긍정단어 리스트에 존재하면
        cnt = cnt +1  #cnt를 1씩 증가시킵니다.
print(cnt)
```

277. emp2.csv에서 이름과 월급을 출력하시오!

```
import csv

file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1], emp_list[5])
```

278. 이름을 물어보게하고 이름을 입력하면 해당 사원의 이름과 월급이 출력되게 하는데 소문자로 입력하던 대문자로 입력하던 잘 출력되게하시오!

이름을 입력하세요~ scott
SCOTT 3000

답:

```
import csv

file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)

a = input('이름을 입력하세요~')

for emp_list in emp_csv:
    if emp_list[1].lower()==a.lower(): #양쪽 다 소문자로 변경
        print(emp_list[1], emp_list[5])
```

279. 스티븐 잡스 연설문에서는 정관사 the가 몇번 나오는가?

```
stev = open("c:₩₩data₩₩jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split()

cnt = 0
for i in stev2:
    if i.lower().strip()== 'the' : #양쪽에 공백이 있을지 모르므로 확인사살 코드
        cnt = cnt +1 #cnt를 1씩 증가시킵니다.
print(cnt)
```

280. 안철수 연설문에서는 국민이라는 단어가 몇번 나오는지 카운트 하시오!

```
ahn = open('c:₩₩data₩₩ahn.txt', encoding = 'UTF8')
ahn2 = ahn.read() #split안했으니까 그냥 안철수 연설문을 읽어서
                  #통째로 ahn2에 입력한다.
print(ahn2) #안철수 연설문 전체가 입력됨
print(ahn2.count('국민')) #안철수 연설문에서 국민이라는 단어가 몇개 있는지 출력하
```

빈다.
#for i in ahn2:
print(i.count('국민'))

281. 동전을 10번 던져서 앞면이 2개가 나오는 지난번 마지막 문제의 확률 10개를 가지고 막대 그래프를 그리시오!

```
import matplotlib.pyplot as plt

y_value = [0.00191, 0.01, 0.07, 0.16]
x_index = [0,1,2,3]
plt.bar(x_index, y_value, color = 'red')
plt.show()
```

282. 문제 281번의 그래파의 결과에 제목, x축라벨과 y축 라벨을 같이 출력하시오.

```
import matplotlib.pyplot as plt

y_value = [0.00191, 0.01, 0.07, 0.16]
x_index = [0,1,2,3]
plt.bar(x_index, y_value, color = 'red')
plt.title('coin Probability') #그래프 제목
plt.xlabel('cnt') # 그래프 x축 라벨
plt.ylabel('probability') #그래프 y축 라벨
plt.show()
```

283. 지난번 마지막 문제인 동전을 10번 던져서 앞면이 2개 나오는 아래의 확률 결과를 가지고 막대그래프로 시각화 하시오!

지난번 마지막 문제의 코드 가져다가 시각화 하세요~

```
def coin_prob(num1):
    import random

    coin = ['앞면', '뒷면']
    cnt = 0

    for k in range(1,10001):
        a = []
        for i in range(1,11):
            result = random.choice(coin)
            a.append(result)

        if a.count('앞면') == num1:
            cnt = cnt +1
    return cnt/10000

import matplotlib.pyplot as plt

y_value = []
x_index = []
for j in range(11):
    y_value.append(coin_prob(j))
    x_index.append(j)


plt.bar(x_index, y_value, color = 'red')
plt.title('coin Probability')
plt.xlabel('tries')
plt.ylabel('probability')
plt.show()
```

284. 우리반 데이터(emp122.csv)를 파이썬으로 로드해서 이메일만 출력하시오.

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[6])
```

285. 위에서 출력한 이메일에서 @의 위치 인덱스 번호를 출력하시오!

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[6].find('@'))
```

286. 그러면 이번에는 이메일에서 .의 위치 인덱스를 출력하시오!

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[6].find('.'))
```

287. 이번에는 이메일을 출력하는데 이메일의 첫번째 철자부터 세번째 철자까지만 출력하시오!

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[6][0:3])
```

288. 우리반 데이터 이메일을 출력하는데 이메일에서 도메인만 출력하시오.

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[6][emp_list[6].find('@')+1:emp_list[6].find('.')])
```

또는

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
        a = emp_list[6].find('@') #@의 위치 인덱스 번호
        b = emp_list[6].find('.') #.의 위치 인덱스 번호
        print(emp_list[6][a+1, b])
```

289. 미승이의 이메일인 miseung.hailey@gmail.com도 같이 출력하려면 어떻게 해야하는가?

```
import csv

file = open('c:₩₩data₩₩emp122.csv', encoding = "UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
        a = emp_list[6].find('@') #@의 위치 인덱스 번호
        b = emp_list[6].rfind('.') #.의 위치 인덱스 번호
        print(emp_list[6][a+1, b])
```

rfind를 하면 뒤에서부터 출력

290. rfind도 domain.co.kr 같은곳은 domain.co까지 끊겨서 문제생기지 않을까요? 의 문제를 해결해보자

```
import csv
file = open("c:₩₩data₩₩emp1222.csv",encoding='UTF8')
emp_csv=csv.reader(file)
for emp_list in emp_csv:
    a = emp_list[6].find('@') #이메일에서 @의 위치 인덱스 번호
    result = emp_list[6][a:]  #@naver.cp.lr <-- @를 포함해서 끝까지 스캔
    b = result.find('.')       #dot(.)의 위치 인덱스 번호
    print(result[1:b])         # @naver.co.kr에서 인덱스번호 1부터 b미만까지
```

291. 아래의 url 변수에 있는 문자열은 슬래쉬(/)로 구분되어있는데 이 문자열의 요소를 아

래의 리스트 처럼 구성하시오!

url = 'http://www.naver.com/news/today=20191204'

결과: ['http:' '', 'www.naver.com', 'news', 'today=20191204']

답:

```
url = 'http://www.naver.com/news/today=20191204'
result = url.split('/')
print(result)
```

292. 우리반 데이터에서 이름만 출력하시오!

```
import csv

file = open('c:₩₩data₩₩emp123.csv', encoding="UTF8")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1])
```

293. 위에서 출력된 이름을 a라는 비어있는 리스트에 담고 a리스트를 출력하시오!

```
import csv

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
emp_csv = csv.reader(file)

a=[]
for emp_list in emp_csv:
    a.append(emp_list[1])

print(a)
```

결과:

['이준혁', '한결', '현지연', '성기창', '유혜영', '김정민', '이성원', '김정원', '김미승', '김예린', '신현종', '구윤모', '김승순', '허정민', '이신성', '황나현', '장수진', '권준환', '송종미', '정희원', '김홍비', '김소라', '권세원', '정다희', '허혁', '양건준', '김주원', '박혜진', '하상준', '홍재연']

294. 위의 a리스트애 담겨진 이름을 하나씩 뽑아서 콤마(,)로 연결해서 아래와 같이 문자열로 출력되게 하시오.

```
import csv

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
emp_csv = csv.reader(file)

a=[]
for emp_list in emp_csv:
    a.append(emp_list[1])
```

```
bond = ','

result = bond.join(a)
print(result)
```

결과:

이준혁,한결,현지연,성기창,유혜영,김정민,이성원,김정원,김미승,김예린,신현종,구윤모,김승순,허정민,이신성,황나현,장수진,권준환,송종미,정희원,김홍비,김소라,권세원,정다희,허혁,양건준,김주원,박혜진,하상준,홍재연

295. 위의 결과가 김씨부터 출력되게 하시오~

(한글을 정렬을 해서 출력하시오 ㄱ ㄴ ㄷ ㄹ 순서로)

```
import csv

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
emp_csv = csv.reader(file)

a=[]
for emp_list in emp_csv:
    a.append(emp_list[1])
a.sort()                        # a 리스트 안의 요소를 정렬을 합니다.
bond = ','

result = bond.join(a)
print(result)
```

296. emp2.csv에서 이름과 월급을 출력하는데 월급을 출력할 때 0 대신에 *로 출력하시오!

SCOTT 3*

SMITH 125*

: :

답:

```
import csv

file = open('c:₩₩data₩₩emp2.csv')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1], emp_list[5].replace('0','*'))
```

297. 아래의 리스트를 가지고 아래와 같이 출력하시오!

a = ['name:홍길동', 'age:17', 'major:경영학', 'nation:한국']

결과:

name ----> 홍길동

age ----> 17

major ----> 경영학
nation---->한국

답:

```
a = ['name:홍길동', 'age:17', 'major:경영학', 'nation:한국']

bond = '₩n'

print(bond.join(a).replace(':', '--->'))
```

298. (점심시간 문제) 우리반 데이터에서 전공을 출력하는데 아래와 같이 나이도 같이 출력되게 하시오!

구윤모(26), 권세원(34), 권준환,김미승,김소라,김승순,김예린,김정민,김정원,김주원,김홍비,박혜진,성기창,송종미,신현종,양건준,유혜영,이성원,이신성,이준혁,장수진,정다희,정희원,하상준,한결,허정민,허혁,현지연,홍재연,황나현

```
import csv

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
emp_csv = csv.reader(file)

a=[]
for emp_list in emp_csv:
    a.append(emp_list[1] +'(' + emp_list[2] + ')')
a.sort() # a 리스트 안의 요소를 정렬을 합니다.
bond = ','

result = bond.join(a)
print(result)
```

299. 아래와 같이 주사위의 눈을 담은 리스트를 만드시오!

print(dice)

결과: [1, 2, 3, 4, 5, 6]

답:

```
dice = list(range(1,7))
print(dice)
```

300. 주사위 2개를 만들고 주사위 2개를 동시에 던져서 두 주사위의 눈의 합이 10이 나오는 확률을 구하시오!

(주사위 2개를 동시에 10000번 던지세요~)

```
import random
dice1 = list(range(1,7))
dice2 = list(range(1,7))

cnt=0
for i in range(1,10001):
```

```
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt += 1
print(cnt/10000)
```

오늘의 문제(301-545)

2020년 12월 9일 수요일 오후 2:06

301. 아래의 결과를 출력하시오!

[2, 4, 6, 8, 10, 12, 14, 16, 18]

```
a = list( range(2, 20, 2) ) #range(시작숫자, 끝숫자, 스텝) 스텝: ~만큼 건너뛰다
print(a)                    # 시작숫자부터 끝숫자 미만까지 스텝만큼 증가해서 출력한다.
```

302. 아래의 리스트에서 숫자 7을 출력하시오!

```
b = [2, 3, 4, [5, 6], [7, 8], 9]
```

답:

```
print(b[4][0])
```

303. 아래의 리스트에서 숫자 4를 출력하시오!

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6] ]
print( a.index(4) )
```

결과:

ValueError: 4 is not in list

답: 리스트 안에 있는 리스트의 인덱스 번호는 출력 못함.

304. list_a의 알파벳 d를 k로 변경하시오!

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
list_a[3] = 'k'
print(list_a)
```

또는

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
list_a[list_a.index('d') ] = 'k'
print(list_a)
```

305. 목차 100번의 예제를 알파벳 a부터 g까지 담은 리스트를 만들었는데 이번에는 알파벳 a부터 z까지를 담는 리스트를 list_a로 생성하시오.

```
import string #string 모듈안에 알파벳이 들어있습니다.

print(string.ascii_lowercase) #abcdefghijklmnopqrstuvwxyz
list_a = []
for i in string.ascii_lowercase:
```

```
        list_a.append(i)
print(list_a)
```

306. 위에서 만든 list_a에서 요소를 검색하는데 맨 끝에 알파벳z빼고 모든 요소를 다 출력하시오!

import string #string 모듈안에 알파벳이 들어있습니다.

#1

```
list_a = []
for i in string.ascii_lowercase:
        list_a.append(i)
print(list_a[:-1])
```

또는

#2

import string #string 모듈안에 알파벳이 들어있습니다.

```
list_a = []
for i in string.ascii_lowercase:
        list_a.append(i)
print(list_a[0 : list_a.index('z')])
```

307. 1부터 100까지의 숫자중에서 짝수만 아래의 list_a에 담아서 출력하시오!

```
print(list_a)
```

[2, 4, 6, 8, 10, 12, ...]

답:

```
list_a = list(range(1,100))
print(list_a[1 : : 2])
```

또는

```
list_a = list(range(2, 101, 2) )
print(list_a)
```

308. 우리반 테이블에서 이름을 출력하는데 ㅎ부터 ㄱ까지 출력될 수 있도록 하시오.

```
import csv

file = open('c:₩₩data₩₩emp1222.csv', encoding="UTF8")
emp_csv = csv.reader(file)

a=[]
for emp_list in emp_csv:
        a.append(emp_list[1] +'(' + emp_list[2] + ')')
a.reverse() # a 리스트 안의 요소를 정렬을 합니다.
bond = ','

result = bond.join(a)
```

```
print(result)
```

309. emp2.csv에서 이름과 월급을 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1], emp_list[5])
```

310. 문제 309번의 데이터중 월급을 비어있는 리스트인 a리스트에 담으시오!

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
a =[]
for emp_list in emp_csv:
    a.append(emp_list[5])
print(a)
```

311. 문제 310번의 a리스트에 있는 월급을 월급이 높은 순서대로 정렬해서 a리스트에 들어 있게 하고 a리스트를 출력하시오.

```
import csv
file = open("c:₩₩data₩₩emp2.csv")
emp_csv = csv.reader(file)
a =[]
for emp_list in emp_csv:
    a.append(int(emp_list[5]))
a.sort() #먼저 낮은 값에서 높은값 순으로 정렬하고
a.reverse() #낮은값에서 높은값 순으로 정렬된 a리스트의 데이터를 역순으로 만듭니다.

print(a)
```

312. 아래와 같이 엄마와 아기가 함께하는 수영교실 나이 리스트를 생성하시오!

결과

[34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

답:

```
listdata1 = [34]
listdata2 = [2]

print(listdata1 * 10 + listdata2*10)
```

313. 아래의 엄마와 아기가 함께하는 수영교실의 나이의 평균값을 출력하시오!

```
swim_age = [34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

```
import numpy as np
a = np.array(swim_age) #numpy array(배열)로 만들어준다.
b = np.mean(a) #numpy로 변환하면 mean을 구할 수 있다.
print(b)
```

314. 아래의 엄마와 아기의 수영교실의 중앙값을 출력하시오!

(중앙값 = 가운데 있는 값)

```
swim_age = [34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

import numpy as np
a = np.array(swim_age) #numpy array(배열)로 만들어준다.
b = np.median(a) #numpy로 변환하면 mean을 구할 수 있다.
print(b) #18.0
```

--> (34 + 2) / 2해서 18이다.

315. 엄마와 아기가 함께하는 수영교실 나이 데이터의 최빈값을 출력하시오.

```
swim_age = [34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

from scipy import stats
import numpy as np
a = np.array(swim_age) #numpy array(배열)로 만들어준다.
result = stats.mode(a)
print(result) #ModeResult(mode=array([2]), count=array([10]))
```

--> 2살이 총 10명있다.

316. 우리반 나이 데이터를 비어있는 리스트 a에 담으시오!

(emp1222.csv로 읽으시오)

```
import csv

file = open("c₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append(int(emp_list[2]))
print(a)
```

317. 우리반 나이 데이터의 평균값과 중앙값과 최빈값을 출력하시오!

```
import csv
from scipy import stats
import numpy as np

file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp_csv = csv.reader(file)
a = []
```

```
for emp_list in emp_csv:
    a.append(np.array(int(emp_list[2])))


print(np.mean(a))
print(np.median(a))
print(stats.mode(a))
```

또는

```
import csv
from scipy import stats
import numpy as np

file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append(int(emp_list[2]))
a2 = np.array(a)
print('평균값:', np.median(a2))
print('중앙값:', np.mean(a2))
print('최빈값:',stats.mode(a2))
```

결과:
평균값: ModeResult(mode=array([28]), count=array([6]))
중앙값: 28.633333333333333
최빈값: 28.0

318. (오늘의 마지막 문제) 우리반 나이 데이터를 가지고 히스토그램 그래프를 그리시오!

x축의 계급(간격) 24~44(2살 간격으로)

```
import csv
import numpy as np
import matplotlib.pyplot as plt

file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp12 = csv.reader(file)
a = []
for emp_list in emp12:
    a.append(int(emp_list[2]))

plt.hist(hist, bins)

bins = list(range(24,45,2))

plt.grid()
plt.hist(a, bins, rwidth = 0.9, alpha = 0.7, color = 'green')
```

- 온라인 Q/A게시판에 올린 문제:

319. 아래의 a리스트의 요소 화성 다음에 소행성을 요소로 입력하시오!
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']

답:
a.insert(5, '소행성')

print(a)

또는
a.insert(a.index('화성')+1, '소행성')

print(a)

320. 아래의 a 리스트에서 목성을 지우시오!
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']
del(a[a.index('목성')])
print(a)

321. 아래의 빨간공 2개와 파란공 3개가 들어있는 주머니에서 공을 하나 랜덤으로 빼면 그 공이 주머니에서 지워지게 하시오.

```
import random

box = ['red', 'blue', 'red', 'blue', 'blue']
ball = random.choice(box)
if result == 'red':
    box.remove('red')
else:
    box.remove('blue')

print(box)
```

또는

```
import random
box = ['red', 'blue', 'red', 'blue', 'blue']
a = random.choice(box)
box.remove(a)
print(box)
```

322. 아래의 주머니에서 임의로 하나의 공을 뽑았을 때 그 공이 파란색일 확률은 어떻게 되는가? (복원추출)

공을 뽑는 작업을 10000번 수행하시오!

```
import random

box = ['red', 'blue', 'red', 'blue', 'blue']

cnt = 0
for i in range(1,10001):
    result = random.choice(box)     #실행문1
    if result =='blue':             #실행문2
        cnt += 1
print(cnt/10000) #0.5983
```

323. 파란공 70개와 빨간공 30개가 들어있는 주머니를 만드시오.

```
a = ['blue']*70
b = ['red']*30
box = a+b
print(box)
```

324. 문제 323번 box에 들어있는 요소가 몇개인지 확인하시오!

```
a = ['blue']*70
b = ['red']*30
box = a+b
print(len(box))  #100
```

325. 위의 box에 있는 요소들을 무작위로 섞으시오!

```
from random import shuffle #from 패키지 import 모듈

a = ['blue']*70
b = ['red']*30
box = a+b
shuffle(box) #box안의 요소들을 무작위로 섞는다.
print(box)
```

326. 위의 box에서 공 3개를 추출했을 때 그 공 3개가 다 파란색일 확률은 어떻게 되는가? (공을 뽑는 작업을 10000번 수행하세요)

```
from random import shuffle #from 패키지 import 모듈
import random

a = ['blue']*70
b = ['red']*30
box = a+b
shuffle(box) #box안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1,10001):
   result1 = random.choice(box)
   result2 = random.choice(box)
   result3 = random.choice(box)
   if result1 == 'blue' and result2 == 'blue' and result3=='blue':
      cnt +=1
print(cnt/10000) #0.3446
```

또는

```
from random import shuffle #from 패키지 import 모듈
import random

a = ['blue']*70
b = ['red']*30
box = a+b
shuffle(box) #box안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1,10001):
    result = random.sample( box, 3) #box에서 공을 3개씩 추출한다.
    if result.count('blue') == 3: #result 리스트의 blue의 개수를 출력
        cnt += 1
print(cnt/10000) #0.3425
```

설명 : random 패키지의 sample 함수는 비복원 추출입니다.

327. 이번에는 위의 box에서 공을 3개를 뽑았을 때 그중 2개가 blue일 확률을 출력하시오!
(복원 추출, 공을 뽑는 작업을 10000번 수행하세요~)

```
from random import shuffle #from 패키지 import 모듈
import random

a = ['blue']*70
b = ['red']*30
box = a+b
shuffle(box) #box안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1,10001):
    result = random.sample( box, 3) #box에서 공을 3개씩 추출한다.
    if result.count('blue') == 2: #result 리스트의 blue의 개수를 출력
        cnt += 1
print(cnt/10000)
```

328. 위의 문제를 복원추출로 수행하시오!

```
from random import shuffle #from 패키지 import 모듈
import random
import numpy as np

a = ['blue']*70
b = ['red']*30
box = a+b
shuffle(box) #box안의 요소들을 무작위로 섞는다.
cnt = 0
cnts = 0
for i in range(1,10001):
    result = list(np.random.choice(box,3, replace = True)) #replace = True면 복원
    cnts +=1
    if result.count('blue')==2:
        cnt +=1
print(cnt/cnts)
```

329. 세원이가 올린 코드를 참조해서 아래의 문제를 해결하시오!

6개의 제품이 들어있는 상자가 있는데 그 중에 2개가 불량품이라고 했을 때 제품 검사를 위해 3개를 추출했을 때 1개가 불량품일 확률을 출력하시오! (복원 추출로 하세요~)

```
box = ['정상', '불량품', '정상', '정상', '불량품', '정상']
shuffle(box) #box안의 요소들을 무작위로 섞는다.
cnt = 0
cnts = 0
for i in range(1,10001):
    result = list(np.random.choice(box,3, replace = True)) #replace = True면 복원
    cnts +=1
    if result.count('불량품')==1:
        cnt +=1
print(cnt/cnts) #0.4429
```

330. a리스트의 요소를 다 지우시오!

```
a = ['태양', '수성','금성', '지구', '화성', '목성', '토성', '천왕성', '헤왕성']
del(a[:]) #인덱스번호 2번부터 6번 미만까지 요소를 지움
print(a) #[]
```

331. (점심시간 문제) 초등학생 10만명의 체중 데이터를 가지고 히스토그램 그래프를 그리시오~
(초등학생 10만명의 평균 체중은 45kg이고 표준편차는 5입니다.)

30~60kg까지 간격 2kg으로 x 축의 간격(계급)을 정하세요~

30~32kg
32~34kg
 : :
58~60kg

```
import numpy as np
import matplotlib.pyplot as plt

weight = np.random.randn(100000)*5 + 45
a = []
for i in weight:
    a.append(int(i))


x_val = list(range(30,61,2))
a = np.array(a)

plt.grid()
plt.xticks(x_val)
plt.hist(a,x_val,rwidth = 0.8, alpha = 0.7, color = 'red')
```

332. 점심시간에 만들었던 초등학생 10만명의 체중 데이터를 모수로 보고 10만명의 체중 데이터에서 100명을 샘플링하여 평균을 구하시오

```
import numpy as np

weight = np.random.randn(100000)*5 + 45
a = []
result = np.random.choice(weight, 100, replace = True).mean()

print(result)
```

333. 문제 332번의 모집단에서 표본 100명을 추출하여 표본평균을 구하는 작업을 1000번 수행해서 1000개의 표본평균을 a라는 비어있는 리스트에 넣으시오

```
import numpy as np

weight = np.random.randn(100000)*5 + 45
a = []
for i in range(1000):
    result = np.random.choice(weight, 100, replace = True).mean()
    a.append(result)

print(a)
```

334. 문제 333번의 a 리스트의 요소의 개수를 출력하시오.

```
import numpy as np

weight = np.random.randn(100000)*5 + 45
a = []
for i in range(1000):
    result = np.random.choice(weight, 100, replace = True).mean()
```

```
    a.append(result)

print(len(a))
```

335. 문제 333번의 표본평균 1000개를 가지고 히스토그램 그래프를 그리시오

x축 계급의 범위: 40~50(간격1)

```
import numpy as np
import matplotlib.pyplot as plt

weight = np.random.randn(100000)*5 + 45
a = []
for i in range(1000):
    result = np.random.choice(weight, 100, replace = True).mean()
    a.append(result)


bins = list(range(40,51,1))
a = np.array(a)

plt.grid()
plt.xticks(bins)
plt.hist(a,bins,rwidth = 0.8, alpha = 0.7, color = 'red')
```

336. 우리반 데이터의 나이를 비어있는 리스트 a에 입력하고 나서 우리반 나이 데이터중에서 27살이 몇명있는지 출력하시오.

```
import csv

file = open("c:₩₩data₩₩emp1222.csv", encoding= 'UTF8')
emp12 = csv.reader(file)

a = []
for emp_list in emp12:
    a.append(int(emp_list[2]))
print(a.count(27)) #5
```

337. 파이썬에서 정의된 모든 변수들을 확인하시오.

```
a = [1,2,3,4,5]
b = 'scott'
c = 30

print(dir())
```

결과:

```
['In', 'Out', '_', '__', '___', '__builtin__', '__builtins__', '__doc__', '__file__', '__loader__',
'__name__', '__package__', '__spec__', '_dh', '_i', '_i1', '_i10', '_i11', '_i12', '_i13', '_i14',
'_i15', '_i16', '_i17', '_i18', '_i19', '_i2', '_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27',
'_i3', '_i4', '_i5', '_i6', '_i7', '_i8', '_i9', '_ih', '_ii', '_iii', '_oh', 'a', 'b', 'bins', 'box', 'c', 'csv',
'emp12', 'emp_list', 'exit', 'file', 'get_ipython', 'i', 'np', 'plt', 'quit', 'result', 'weight']
```

지우개로 지우고

print(dir())
하니
['In', 'Out', '__builtin__', '__builtins__', '__file__', '__name__', '_dh', '_i', '_i28', '_ih', '_ii', '_iii', '_oh', 'exit', 'get_ipython', 'quit']

이렇게 나옴.

338. emp2.csv 에서 월급을 비어있는 리스트인 a에 담고 월급을 역순으로 정렬해서 리스트의 요소로 구성하시오.

```
import csv

file = open("c:₩₩data₩₩emp2.csv")
emp = csv.reader(file)
a = []
for emp_list in emp:
    a.append(int(emp_list[5]))

a.sort(reverse=True)

print(a)
```

339. 아래의 artist라는 리스트에 가수이름들을 추가하고 music리스트에 음악리스트를 추가하시오!

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

340. 문제 339번의 artist를 키로 하고 music을 값으로 해서 아래와 같은 딕셔너리를 구성하시오!

artist = ['비틀즈', '아이유', '마이클 잭슨']
music = ['yesterday','너랑나', 'beat it']

gini = {'비틀즈': 'yesterday', '아이유':'너랑나', '마이클잭슨':'beat it'}

답:

```
gini = {}
gini['비틀즈'] = 'yesterday'
        ↑              ↑
        키             값
ginin['아이유'] = '너랑나'
gini['마이클 잭슨'] = 'beat it'
print(gini)
```

최종:

{'비틀즈': ['yesterday', 'imagine',] '아이유': ['너랑나', '마슈멜로우'] , '마이클잭슨':

['beat it', 'smooth criminal']}

341. 문제 340번 같이 일일이 손으로 노가다 하지 말고 아래의 2개의 리스트를 이용해서 for문과 zip을 이용해서 한번에 gini 딕셔너리를 구성하시오.

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
music = ['yesterday','너랑나', 'beat it']
gini = {}

for i,k in zip(artist, music):
    gini[i] = k
print(gini)
```

342. 아래의 music 리스트에서 요소를 하나씩 빼내는데 앞에 인덱스 번호도 같이 출력되게 하시오.

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
```

결과:

```
0 yesterday
1 imagine
2 너랑나
3 마슈멜로우
4 beat it
5 smooth criminal
```

답:

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

for i, k in enumerate(music):
    print(i,k)
```

343. artist리스트를 가지고 아래와 같이 결과를 출력하시오!

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

결과:

```
비틀즈 0
비틀즈 1
아이유 2
아이유 3
마이클 잭슨 4
마이클 잭슨 5
```

답:

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

```
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

for i, k in enumerate(artist):
    print(k,i)
```

344. 아래의 두개의 리스트를 가지고 아래와 같이 결과를 출력하시오!

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
(결과)
비틀즈 yesterday
비틀즈 imagine
아이유 너랑나
아이유 마슈멜로우
마이클 잭슨 beat it
마이클 잭슨 smooth criminal

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

for i, k in enumerate(artist):
    print(k, music[i])
```

최종

{'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'],
↑ ↑
키 값
'마이클잭슨': ['beat it', 'smooth criminal']}

설명: 위와 같이 딕셔너리를 구성하려면 그냥 딕셔너리인 중괄로 하나로 { } 비어있는 딕셔너리를 만들면 안되고 defaultdict를 이용해야 합니다. defaultdict는 collections 패키지에 있어서 아래와 같이 임폴트 해야합니다.

```
from collections import defaultdict

gini = defaultdict(list)
gini['비틀즈'].append('yesterday')
print(gini)
gini['비틀즈'].append('imagine')
print(gini)
```

결과:

```
defaultdict(<class 'list'>, {'비틀즈': ['yesterday']})
defaultdict(<class 'list'>, {'비틀즈': ['yesterday', 'imagine']})
```

345. 그럼 문제 344번에서 만든 default 딕셔너리에 아이유를 키로 하고 너랑나, 마슈멜로

우를 값으로 해서 추가하시오!

```
from collections import defaultdict

gini = defaultdict(list)
gini['비틀즈'].append('yesterday')
gini['비틀즈'].append('imagine')
gini['아이유'].append('너랑나')
gini['아이유'].append('마슈멜로우')
gini['마이클잭슨'].append('beat it')
gini['마이클잭슨'].append('smooth criminal')

print(gini)
```

346. 문제 345번과 같이 일일이 손으로 노가다 하지말고 enumerate를 이용해서 gini 딕셔너리를 만드시오.

```
from collections import defaultdict

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])
print(gini)
```

결과:

```
defaultdict(<class 'list'>, {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']})
```

347. 문제 346번의 gini 딕셔너리에서 음악만 추출하시오. (딕셔너리의 값만 추출)

딕셔너리에서 키값을 추출할 때는: 딕셔너리이름.keys( )
딕셔너리에서 값만 추출할 때는 : 딕셔너리이름.values( )

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])
print(gini)
for i in gini.values( ):
    print(i)
```

결과:

```
['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']
```

```
------------------
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])
for i in gini.values( ):
        for k in i:  #['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth
        criminal']
                print(k)
```

348. 위의 코드 음악들을 비어있는 리스트인 a에 담으시오!

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])
a = []
for i in gini.values( ):
        for k in i:  #['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth
        criminal']
                a.append(k)
print(a)
```

349. 아래의 코드를 수행해서 gini.values()의 요소들을 프린트 해보시오!

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])
for j in gini.values():
        print(j)
```

결과:

```
['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']
------------------------------------------
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
        gini[k].append(music[i])
for j in gini.values():
```

```
    print(j[0])
```

결과:
yesterday
너랑나
beat it

350. (오늘의 마지막 문제 12/10) 아래와 같이 결과를 출력하시오!
노래가 나와줘야 합니다.
yesterday, 너랑나, beat it, imagine, 마슈멜로우, smooth criminal

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

gini = defaultdict(list)
for i, k in enumerate(artist):
    gini[k].append(music[i])



a = []
b= []
for j in gini.values():
    a.append(j[0])
    b.append(j[1])
bond = ','
result = bond.join(a+b)
print(result)
```

best답변 by 한결님:

```
from collections import defaultdict

gini = defaultdict(list)

artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']

music = ['yesterday','imagine','너랑나','마쉬멜로우','beat it','smooth criminal']

a = []

for i, k in enumerate(artist):

    gini[k].append(music[i])



for q in range(2):

    for j in gini.values():
```

```
        a.append(j[q])

print(', '.join(a))
```

351. 아래의 리스트의 짝수번째 요소의 합을 출력하시오!
listdata = [2, 2, 1, 3, 8, 4, 3, 9, 2, 20]

답:

```
listdata = [ 2, 2, 1, 3, 8, 4, 3, 9, 2, 20]
result = sum(listdata[1 : : 2])
print(result)
```

352. (알고리즘 문제) 아래의 리스트에서 1부터 10까지의 숫자중에 없는 숫자가 하나있다. 그 없는 숫자는 무엇인가?
a = [2, 1, 5, 4, 6, 7, 9, 10, 3]

답:

```
a = [2, 1, 5, 4, 6, 7, 9, 10, 3]

result = sum(range(1,11))- sum(a)
print(result)
```

353. while loop문을 이용해서 무한루프문을 수행하시오!

```
while True:
      print('aaaaaaaaaaaaaa')
```

무한대로 계속 실행이되어서 종료하려면 오른쪽의 빨간 정지버튼을 눌러줘야한다.

354. 이번에는 문제 353번의 True대신에 숫자 1을 넣고 무한 루프문을 수행하시오!

```
while 1:
   print('aaaaaaaaaaaaaa')
```

355. 아래의 2개의 리스트를 가지고 sol 딕셔너리를 생성하시오!

```
sol_eng = ['sun', 'mercury', 'venus', 'earth', 'mars']
sol_kor = ['태양', '수성', '금성', '지구', '화성']
```

결과:

```
print(sol)
# {'태양': 'sun', '수성': 'mercury', '금성': 'venus', 'sun': '태양', 'mercury': '수성', 'venus':
'금성', 'earth': '지구', 'mars': '화성'}
```

답:

```
sol_eng = ['sun', 'mercury', 'venus', 'earth', 'mars']
sol_kor = ['태양', '수성', '금성', '지구', '화성']
```

```
for i, k in zip(sol_eng, sol_kor):
    sol[i] = k
print(sol)
```

356. 아래의 딕셔너리의 값중에 Fire를 피땀눈물로 변경하시오!

```
dict = {'방탄소년단':'Fire', '소녀시대':'Gee'}
print(dict) #{'방탄소년단': 'Fire', '소녀시대': 'Gee'}
dict['방탄소년단''] = '피땀눈물'
print(dict) #{'방탄소년단': '피땀눈물', '소녀시대': 'Gee'}
```

357. 아래의 딕셔너리의 값중에서 Fire를 피땀눈물로 변경하시오!
dict = {'소녀시대' : ['다시만난세계', 'Gee'], '방탄소년단' : ['DNA', 'Fire']}

답:

```
dict = {'소녀시대' : ['다시만난세계', 'Gee'], '방탄소년단' : ['DNA', 'Fire']}

dict['방탄소년단'][1] = '피땀눈물'
print(dict)
```

358. 아래의 딕셔너리에서 다시만난세계의 값만 지우시오!
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}

답:

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
del dict['소녀시대'][0]
print(dict)
```

359. 목자 123번 예제와 같이 딕셔너리의 요소를 지우는게 아니라 딕셔너리 자체를 메모리에서 지우려면 어떻게 해야하는가?

목차 123번 예제:

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
dict.clear()
print(dict)
```

답:

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
del dict
print(dict)   #<class 'dict'>
```

360. 다음에서 sol사전에 있는 키를 추출해서 아래와 같이 결과를 출력하시오.
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
결과:
태양,수성,금성,지구,화성

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
a = []
for i in sol.keys():
    print(i, end = ',') #태양,수성,금성,지구,화성,
```

설명: print의 end옵션에 준 구분자로 값들을 가로로 출력하면서 구분자로 구분해줍니다.

361. 문제 361번의 결과에서 맨 끝에 있는 콤마(,)는 출력되지 않게 하시오!
결과 : 태양,수성,금성,지구,화성

362. 아래의 sol 딕셔너리에서 값들만 아래와 같이 출력하시오!
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
결과: sun,mercury,venus,earth,mars,

답:
```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구':'earth', '화성':'mars'}
for i in sol.values():
    print(i, end=',')
```

363. 아래의 ginin 딕셔너리에서 값만 추출하시오!
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

결과:
['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']

답:
```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

for i in gini.values():
    print(i)
```

364. 문제 363번의 결과를 다시 출력하는데 결과 리스트 3개중에 0번째 요소만 출력하시오!

결과:
yesterday
너랑나
beat it

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```
for i in gini.values():
    print(i[0])
```

365. 지난주 목요일 마지막 문제의 자신의 답을 가져와서 gini 딕셔너리에서 음악만 출력하는데 아티스트별로 노래가 겹치지 않게 출력하시오!

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

a = []
for j in gini.values():
        a.append(j[0])
for j in gini.values():
        a.append(j[1])

bond = ','
result = bond.join(a)
print(result)
```

366. 아래의 gini 딕셔너리의 값들의 리스트를 아래의 결과처럼 리스트에 담으시오!
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

결과:
[['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth criminal']]

답:

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

gini2 = list(gini.values()) #리스트형으로 변환
print(gini2)
```

367. 문제 366번에서 출력된 gini2 리스트의 요소들을 shuffle로 섞어 보시오!

```
from random import shuffle

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

gini2 = list(gini.values()) #리스트형으로 변환
shuffle(gini2)
print(gini2)
```

368. 지난시간 만들었던 마지막 문제의 자신의 답과 리스트의 요소를 섞는 shuffle함수를 이용해서 코드를 수행할 때마다 곡이 무작위로 섞여서 출력되게 하시오! (단 조건은 아티스트별로 노래가 겹치면 안된다. 음악 바로 옆에 다른 아티스트의 노래가 나와야 합니다.

369. 아래 gini딕셔너리의 요소들의 items를 다시 뽑아서 리스트에 담으시오

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
print(gini.items())
```

답:

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
result =list(gini.items())
print(result)
```

결과: [('비틀즈', ['yesterday', 'imagine']), ('아이유', ['너랑나', '마슈멜로우']), ('마이클 잭슨', ['beat it', 'smooth criminal'])]

370. 문제 369번의 result에서 음악 첫번째 곡들만 아래와 같이 출력하시오!

결과:

yesterday
너랑나
beat it

답:

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
result =list(gini.items())
for i in result:
    print(i[1][0])
```

371. 다음 결과를 reverse 되게 해서 출력하시오!

```
dict2 = {'소녀시대':'소원을 말해봐', '방탄 소년단':'DNA', '오마이걸':'살짝 설렜어'}
result = dict2.keys()
print(sorted(result)) #['방탄 소년단', '소녀시대', '오마이걸']
```

답:

```
dict2 = {'소녀시대':'소원을 말해봐', '방탄 소년단':'DNA', '오마이걸':'살짝 설렜어'}
result = dict2.keys()
print(sorted(result, reverse = True)) #['오마이걸', '소녀시대', '방탄 소년단']
```

372. 우리반 데이터(emp1222.csv) 에서 이름과 통신사를 출력하시오!

```
import csv
file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp = csv.reader(file)

for emp_list in emp:
    print(emp_list[1], emp_list[5])
```

결과:

이준혁 kt
한결 kt
현지연 sk
성기창 sk
유혜영 sk
김정민 lg
이성원 kt
김정원 kt
김미승 lg
김예린 kt
신현종 lg
구윤모 sk
김승순 sk
허정민 sk
이신성 sk
황나현 kt
장수진 kt
권준환 kt
송종미 kt
정희원 sk
김홍비 kt
김소라 kt
권세원 sk
정다희 kt
허혁 kt
양건준 sk
김주원 lg
박혜진 sk
하상준 kt
홍재연 kt

373. 우리반 데이터에서 통신사를 key로 하고 학생이름을 값으로 해서 defaultdictionary를 생성하시오!

(defaultdictionary 이름은 telecom으로 하세요)

```
from collections import defaultdict
telecom = defaultdict(list)


file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp = csv.reader(file)

for emp_list in emp:
    telecom[ emp_list[5]].append(emp_list[1])

print(telecom)
```

결과:

defaultdict(<class 'list'>, {'kt': ['이준혁', '한결', '이성원', '김정원', '김예린', '황나현', '장수진', '권준환', '송종미', '김홍비', '김소라', '정다희', '허혁', '하상준', '홍재연'], 'sk': ['현지연', '성기창', '유혜영', '구윤모', '김승순', '허정민', '이신성', '정희원', '권세원', '양건

준', '박혜진'], 'lg': ['김정민', '김미승', '신현종', '김주원']})

374. 문제 373번에서 구성한 telecom 딕셔너리에서 통신사가 kt인 학생들을 출력하는데 이름을 ㄱ, ㄴ, ㄷ, ㄹ순으로 정렬해서 출력하시오!

```
from collections import defaultdict
file = open("c:₩₩data₩₩emp1222.csv", encoding = "UTF8")
emp = csv.reader(file)

telecom = defaultdict(list)

for emp_list in emp:
    telecom[ emp_list[5]].append(emp_list[1])

result = telecom['kt']
print(sorted(result)
```

375. 알파벳 대문자를 출력하시오!

```
import string
print(string.ascii_uppercase)
```

결과:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

376. 문제 375번의 코드와 ord()함수를 이용해서 아래의 결과를 출력하시오!

결과: A ----------> 65
B ----------> 66
:
Z ---------->

```
import string

for i in string.ascii_uppercase:
    print(i, '---------->' , ord(i))
```

377. 아래와 같이 결과를 출력하시오!

결과:

65 ----------> A
66 ----------> B
:
90 ----------> Z

```
for i in range(65, 91):
    print(i, '---------->', chr(i))
```

378. 아래의 결과를 출력하시오!

34 + 256 -71*34 = ?

답:

```
a = '34 + 256 -71 * 34'
print(a, ' = ', eval(a))
```

379. 아래의 리스트의 숫자를 뽑아내서 아래의 문자열을 생성하시오!

```
a = [34, 22, 45, 27, 31, 33, 55]
b = []

for i in a:
    b.append(str(i))

print(b)     # ['34', '22', '45', '27', '31', '33', '55']
bond = '+'
result = bond.join(b)
print(result) #34+22+45+27+31+33+55
```

380. 아래의 리스트를 가지고 아래의 결과를 출력하시오!

```
a = [34, 22, 45, 27, 31, 33, 55]
b = []

for i in a:
    b.append(str(i))

result = ' + '.join(b)
print(result, '=', eval(result)) #34 + 22 + 45 + 27 + 31 + 33 + 55 = 247
```

381. 우리반 데이터(emp1222.csv)에서 나이를 읽어들여 아래와 같이 결과가 출력되게 하시오.

결과 : 24 + 27 + 26 + 27 +..... 44= ?.

답:

```
import csv

file = open("c:₩₩data₩₩emp1222.csv", encoding="UTF8")
emp = csv.reader(file)

a = []
for emp_list in emp:
    a.append(str(emp_list[2]))

result = ' + '.join(a)
print(result, '=', eval(result))
```

결과:

29 + 31 + 35 + 29 + 28 + 28 + 25 + 28 + 27 + 27 + 27 + 26 + 33 + 30 + 27 +

27 + 25 + 28 + 29 + 24 + 24 + 29 + 36 + 26 + 26 + 26 + 44 + 28 + 29 + 28 = 859

382. 아래의 함수를 그냥 이름없는 한줄짜리 함수로 만들어 보시오!

```
def gob_func(a,b):
    return a*b
```

답:

```
k = lambda a,b : a*b
print( k(2,3) )
```

383. 아래의 함수를 그냥 이름 없는 한줄짜리 함수로 생성하시오!

```
def odd_func(a):
    if a % 2 == 0:
        return '짝수 입니다'
    else:
        return '홀수 입니다'
```

답:

```
k = lambda a: '짝수 입니다' if a % 2 == 0 else '홀수입니다' #조건 먼저!!
print(k(2))
print(k(3))
```

384. 아래의 함수를 lambda로 구현하시오!

```
def high_income(a):
    if a >= 3000:
        return '고소득자 입니다'
    else:
        return '일반 소득자 입니다'
```

답:

```
k = lambda a: '고소득자 입니다' if a >=3000 else '일반 소득자 입니다'
print( k(4000))
```

385. 아래의 함수를 lambda로 구현하시오!

```
def child_tall(num):
    if 140-1.96*5 < = num <= 140+1.96*5:
        return '신뢰구간 95% 안에 있습니다'
    else:
        return '신뢰구간 95% 안에 없습니다'
```

신뢰구간 68% -------> 평균 ± 1 + 표준편차

신뢰구간 95% -------> 평균 ± 1.96 + 표준편차

신뢰구간 99% -------> 평균 ± 2.58 + 표준편차

답: 한줄짜리 함수로는 너무 길어서 lambda로 하지 않는 것이 좋다.

위에서 만든 def 함수 확인하기

```
def child_tall(num):
    if 140-1.96*5 <= num <= 140+1.96*5:
        return '신뢰구간 95% 안에 있습니다'
    else:
        return '신뢰구간 95% 안에 없습니다'

print(child_tall(147)) #신뢰구간 95% 안에 있습니다
print(child_tall(157)) #신뢰구간 95% 안에 없습니다
```

386. 초등학생 키 데이터를 120~160 사이로 해서 0.001 간격으로 생성하시오!

```
import numpy as np
x = np.arange(120, 160, 0.001)
print(len(x)) #40000
```

387. 문제 386번에서 만든 40000건의 데이터에 대한 확률밀도 함수값을 출력하시오! (평균값 140, 표준편차 5)

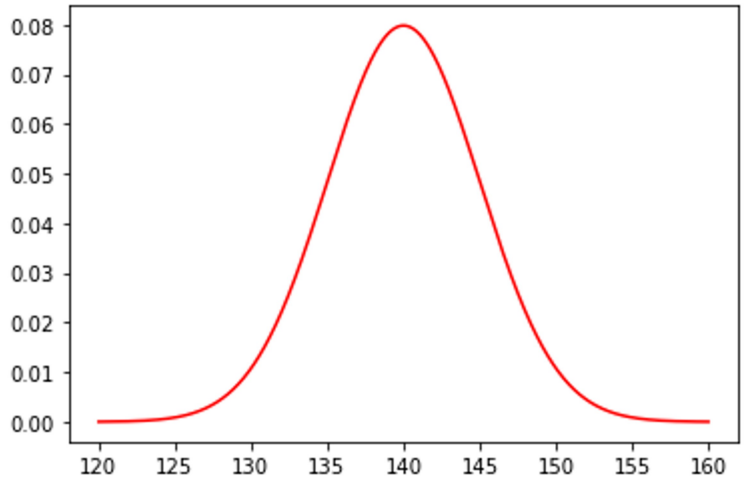
```
from scipy.stats import norm
import numpy as np

x = np.arange(120, 160, 0.001)
y = norm.pdf(x, 140, 5) #확률밀도함수
print(y)
```

결과:

```
[2.67660452e-05 2.67874660e-05 2.68089030e-05 ... 2.68303560e-05
 2.68089030e-05 2.67874660e-05]
```

388. 문제 387번의 데이터 중 x값과 y값을 가지고 시각화 하시오!

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(120, 160, 0.001)
y = norm.pdf(x, 140, 5)

plt.plot(x, y, color = 'red')
plt.show()
```



389. 문제 388번의 그래프에서 신뢰구간 95%에 해당하는 부분만 색깔로 칠하시오.

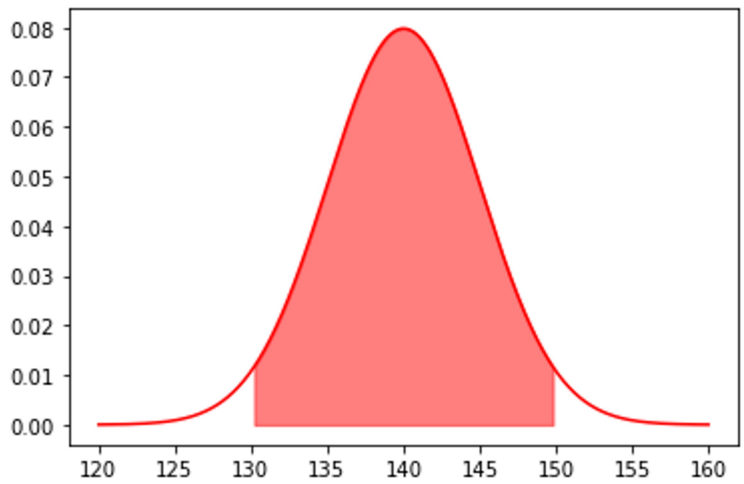
신뢰구간 68% -------> 평균 ± 1 + 표준편차

신뢰구간 95% -------> 평균 ± 1.96 + 표준편차

신뢰구간 99% -------> 평균 ± 2.58 + 표준편차

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(120, 160, 0.001)
y = norm.pdf(x, 140, 5)

plt.plot(x, y, color = 'red')
plt.fill_between( x, y, where = (x>=140 - 1.96*5) & (x<= 140 + 1.96*5), color= 'red',
alpha = 0.5)
plt.show()
```

설명: where에 색칠하는 구간을 지정합니다.



390. (오늘의 마지막 문제 12/14) 위의 코드를 이용해서 confidence_interval 함수를 생성하고 아래와 같이 신뢰구간을 넣으면 해당 신뢰구간에 해당하는 그래프가 출력되게 하

시오.

신뢰구간 68% -------> 평균 ± 1 + 표준편차

신뢰구간 95% -------> 평균 ± 1.96 + 표준편차

신뢰구간 99% -------> 평균 ± 2.58 + 표준편차

confidence_interval(68)
confidence_interval(95)
confidence_interval(99)

답:

```
def confidence_interval(num):
    import matplotlib.pyplot as plt
    from scipy.stats import norm
    import numpy as np

    if num == 68:
        k = 1
    elif num == 95:
        k = 1.96
    elif num == 99:
        k = 2.58

    x = np.arange(120, 160, 0.001)
    y = norm.pdf(x, 140, 5)

    plt.plot(x, y, color = 'red')
    plt.fill_between( x, y, where = (x>=140 - k*5) & (x<= 140 + k*5), color= 'red',
alpha = 0.5)
    plt.show()


print(confidence_interval(68))
```

```
print(confidence_interval(95))
```

print(confidence_interval(99))

391. 숫자를 입력해서 해당 숫자가 3000보다 크면 숫자 1을 리턴하게 하고 입력한 숫자가 3000보다 작다면 0을 리턴하는 함수를 생성하시오.

print(high_income(3500))

결과: 1

print(high_income(2000))

결과 : 0

답:

```
def high_income(num):
    if num >= 3000:
        return 1
    else:
        return 0

print(high_income(3500)) #1
```

392. 문제 391번에서 만든 함수를 map함수에 적용해서 아래의 리스트의 요소를 high_income 함수의 입력 매개변수로 입력해서 아래의 결과가 출력되게 하시오.

```
def high_income(num):
    if num >= 3000:
        return 1
    else:
        return 0

a = [4000, 5000, 2000, 3500 , 1000]


result = map(high_income, a)
print( list(result) ) #[1, 1, 0, 1, 0]
```

393. 동전을 지정된 숫자만큼 던져서 앞면이 나오는지 뒷면이 나오는지 출력하는 함수를 생성하시오.

```
def coin_cnt(num):
    import random
    coin = ['앞면', '뒷면']
    for i in range(1, num+1):
        result = random.choice(coin)
        print (result)

coin_cnt(5)
```

394. 문제 393번의 함수를 수정해서 숫자를 입력하면 해당 숫자만큼 동전을 던져서 동전이 앞면이 나오는 확률을 출력하시오.

```
def coin_cnt(num):
    import random
    coin = ['앞면', '뒷면']
    cnt = 0
    for i in range(1, num+1):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt+1
    return cnt/num

print(coin_cnt(100)) #0.54
```

395. 문제 394번에서 만든 coin_cnt 함수에 아래의 a리스트의 요소들을 적용해서 결과로 확률이 출력되게 하시오.
a = [10, 100, 1000, 10000, 100000]

결과:
[0.6, 0.58, 0.506, 0.4979, 0.4986]

답:

```
def coin_cnt(num):
    import random
    coin = ['앞면', '뒷면']
    cnt = 0
    for i in range(1, num+1):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt+1
    return cnt/num

a = [10, 100, 1000, 10000, 100000]

k = map(coin_cnt, a)
print(list(k))
```

설명 : 동전을 던지는 횟수가 많아질수록 0.5에 근사한 값이 출력되고 있다.

396. 주사위를 던져서 주사위의 눈이 5가 나올 확률을 출력하는 함수를 만들고 아래의 a리스트의 주사위를 던지는 횟수를 map함수로 적용해서 확률이 점점 1/6로 근사해지는지 실험히사오!

```
a = [10, 100, 1000, 10000, 100000]
result = map(dice_cnt, a)
print(list(result))
```

딥:

```
a = [10, 100, 1000, 10000, 100000]
def dice_cnt(num):
    import random
    cnt = 0
    dice = [list(range(1,7))
    for i in range(1, num+1):
        result = random.choice(dice)
        if result == 5:
```

```
            cnt += 1
    return cnt/num

k = map(dice_cnt, a)
print(list(k))
```

397. 아래의 불량품이 들어있는 박스에서 제품을 3개를 뽑았을 때 3개중에서 2개가 불량품일 확률을 출력하는 함수를 만드시오~

함수를 실행할 때 3개를 뽑은 횟수를 입력매개변수로 사용하세요~

```
box = ['정상', '정싱', '불량품', '정상', '불량품', '정상', '정상', '불량품']
print( box_cnt(100) )

def box_cnt(num):
    import numpy as np
    box = ['정상', '정싱', '불량품', '정상', '불량품', '정상', '정상', '불량품']
    cnt = 0
    for i in range(1, num+1):
        result = list(np.random.choice(box, 3, replace = True))
        if result.count('불량품') == 2:
            cnt +=1
    return cnt/num

print(box_cnt(100))
```

398. 위에서 만든 box_cnt 함수에 아래의 a리스트의 횟수를 map함수로 적용해서 확률이 출력되게 하시오!

```
def box_cnt(num):
    import numpy as np
    box = ['정상', '정싱', '불량품', '정상', '불량품', '정상', '정상', '불량품']
    cnt = 0
    for i in range(1, num+1):
        result = list(np.random.choice(box, 3, replace = True))
        if result.count('불량품') == 2:
            cnt +=1
    return cnt/num

a = [10, 100, 1000, 10000, 100000]

result = map(box_cnt, a)
print(list(result))
```

399. 중앙일보에서 인공지능으로 검색한 기사인 mydata3.txt를 파이썬으로 불러서 출력하시오!

```
f = open("c:₩₩data₩₩mydata3.txt")
data = f.read()
print(data)
f.close() #파일을 닫는다. 담는 코드를 안쓰면 스파이더를 실행하고 있으면
          #계속 파일이 열려있게 됩니다. 파일을 계속 열면 메모리를 계속 차지하며 사용하고 있게됩니다.
          #텍스트 파일의 크기가 크면 메모리를 많이 사용하고 있게 됩니다.
          #그래서 사용이 끝났으면 반드시 close()로 닫습니다.
```

400. 위의 중앙일보 기사에서 빅데이터라는 단어가 몇번 나오는지 카운트 하시오.

```
f = open("c:₩₩data₩₩mydata3.txt", encoding = 'UTF8')
data = f.read()
print(data.count('빅데이터')) #11번
f.close()
```

401. 스티븐 잡스 연설문 데이터를 모두 읽어 오시오.
(readline()함수를 이용하세요~)

```
f = open("c:₩₩data₩₩jobs.txt")
data = f.readline()    #while문이 실행될 수 있도록 마중물 역할을 한다.
while data:            #data 변수안에 data가 있으면 True 없으면 False입니다.
    print(data)
    data = f.readline() #그 다음줄을 읽어서 data변수에 입력합니다.
f.close()
```

설명 : 맨 위의 data = f.readline() 코드는 처음에 딱 한번만 실행되고 그 다음에는 실행되지 않습니다. 근데 이게 없으면 while문이 실행이 안된다. data가 들어와야지 while문 실행
그 다음부터는 while문 안에 있는 data = f.readline()가 작동되어서 반복적으로 스티븐 잡스의 연설문을 한줄씩 읽어서 data변수에 넣습니다.

402. 401번에서 한줄씩 읽은 데이터에서 인공지능이라는 단어가 count되게 하시오.

```
f = open("c:₩₩data₩₩mydata3.txt", encoding= ('UTF8'))
data = f.readline()    #while문이 실행될 수 있도록 마중물 역할을 한다.
while data:            #data 변수안에 data가 있으면 True 없으면 False입니다.
      print(data.count('인공지능'))
      data = f.readline() #그 다음줄을 읽어서 data변수에 입력합니다.
f.close()
```

403. mydata3.txt의 count한 건수를 다 누적시켜서 출력하시오. (readline 사용)

```
f = open("c:₩₩data₩₩mydata3.txt", encoding= ('UTF8'))
data = f.readline()
cnt = 0    #while문이 실행될 수 있도록 마중물 역할을 한다.
while data:            #data 변수안에 data가 있으면 True 없으면 False입니다.
    cnt += data.count('인공지능')
    data = f.readline() #그 다음줄을 읽어서 data변수에 입력합니다.
print(cnt)
f.close()
```

404. 위의 코드를 수정해서 단어를 물어보게하고 단어를 입력하면 해당 단어가 몇건 나오는지 출력되게 하시오.

단어를 입력하세요~ 입력지능

이 기사에서 인공지능은 24번 나왔습니다.

답:

```
f = open("c:₩₩data₩₩mydata3.txt", encoding= ('UTF8'))
data = f.readline()
cnt = 0    #while문이 실행될 수 있도록 마중물 역할을 한다.
word = input('단어를 입력하세요~ ')
while data:            #data 변수안에 data가 있으면 True 없으면 False입니다.
    cnt += data.count(word)
    data = f.readline() #그 다음줄을 읽어서 data변수에 입력합니다.
print('이기사에서',word,'단어는 ', cnt,'번 나왔습니다.')
f.close()
```

405. (점심시간 문제) 아래의 파란색 공과 빨간색 공이 들어있는 박스에서 5개의 공을 뽑았을 때 그 중 2개가 파란색 공일 확률을 출력하는 함수를 만들고 공을 뽑는 횟수를 10, 100, 1000, 10000, 100000번 했을때의 확률을 map함수를 이용해서 출력하시오.

a = [10, 100, 1000, 10000, 100000]

파란색 공: 24개 + 빨간색공 26개

답:

```
def box_cnt(num):
    import numpy as np
    box = ['파란공']*24 + ['빨간공']*26
    cnt = 0
    for i in range(1, num+1):
        result = list(np.random.choice(box, 5, replace = True))
        if result.count('파란공') == 2:
            cnt += 1
```

```
    return cnt/num

a = [10, 100, 1000, 10000, 100000]

m = map(box_cnt, a)

print(list(m))
```

406. 아래의 리스트를 mydata9.txt 로 저장하시오!
listdata2 = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

답:

```
listdata2 = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
result = sorted( listdata2 )
f = open("c:₩₩data₩₩mydata9.txt", "w") #mydata2.txt를 생성하겠다.
f.write( str(result) ) #result에 있는 내용을 mydata2.txt로 생성한다.
f.close()
```

407. 다음과 같이 내용과 링크를 수정하시오~
(링크는 아무거나하세요~)

양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육탕으 먹었습니다. 코로나에대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이지는 바로가기를 누르세요. 바로가기

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다.
</i></u></b></p>
<p class="content">양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육탕으 먹었습니다. 코로나에대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이지는 바로가기를 누르세요.

<a href="https://www.google.com/search?q=%EC%9A%B0%EC%9C%A1%ED%83%95&rlz=1C1CHZN_koKR924KR924&oq=%EC%9A%B0%EC%9C%A1%ED%83%95&aqs=chrome..69i57j0l7.4906j0j7&sourceid=chrome&ie=UTF-8" class="cafe1" id="link1"> 바로가기 </a>
</p>
</body>
</html>
```

설명: 워드 문서를 만들때도 — 문서내에 단원도 있고 세부 목차도 있듯이 class에 그 html문서의 특정 단원이라고 보면 되고 id는 링크를 줄 때 부여하는 제목인데 id는 값이 unique합니다.

408. ecologicalpyramid.html 문서에서 number클래스에 있는 숫자들만 긁어와서 출력하시오!

```
f = open("c:₩₩data₩₩ecologicalpyramid.html")
soup = BeautifulSoup( f, "html.parser")
result = soup.find_all(class_ ="number")
for i in result: #result 리스트안의 요소를 하나씩 가져옵니다
    print(i.get_text())
```

결과:
100000
100000
1000
2000
100
100
80
50

409. 문제 408번에서 긁어온 숫자들을 a라는 비어있는 리스트에 저장한 후 a안의 요소들을 정렬하고 a 리스트를 출력하시오!

```
f = open("c:₩₩data₩₩ecologicalpyramid.html")
soup = BeautifulSoup( f, "html.parser")
result = soup.find_all(class_ ="number")
a = []
for i in result: #result 리스트안의 요소를 하나씩 가져옵니다
    a.append(int(i.get_text()))  #이렇게 해줘야 제대로 정렬이 된다.
a.sort()
print(a)
```

결과:
[50, 80, 100, 100, 1000, 2000, 100000, 100000]

※ a.sort()를 따로 안쓰고 print(a.sort())라고 바로 써주면 sort하고 있는 과정에 프린트를 하는거라서 이렇게 하면 none이 나온다

410. 중앙일보사 홈페이지로 가서 신문기사 하나를 보시오.

ctrl + s로 기사의 html문서를 aa77.html로 저장합니다.

411. aa77.html을 beautifulsoup 모듈의 함수를 쓸 수 있도록 파싱하고 파싱된 결과를 출력하세요~

```
from bs4 import BeautifulSoup
f = open("c:₩₩data₩₩aa77.html", encoding='UTF8')
soup = BeautifulSoup( f, "html.parser")
print(soup)
```

412. 위의 기사의 본문을 가져오기 위해서 기사 본문의 class이름이 무엇인지 확인하세요

크롬의 개발자 모드 단축키인 F12를 누르면 옆에 html 코드 창이 나오면서 개발자 모드 화면이 열립니다. 그러면 위쪽에 화살표를 클릭하고 기사본문을 클릭하며 html문서에 class이름을 확인할 수 있는 html문서 쪽으로 바로 이동합니다.

클래스 이름: article_body mg fs4

413. 클래스 이름 article_body mg fs4로 접근하여 text를 긁어오시오.

```
from bs4 import BeautifulSoup
f = open("c:₩₩data₩₩aa77.html", encoding='UTF8')
soup = BeautifulSoup( f, "html.parser")

result = soup.find_all(class_ = 'article_body mg fs4')
for i in result:
    print(i.get_text())
```

414. (오늘의 마지막 문제 12/15) 문제 413번에서 스크롤링한 중앙일보 기사를 c드라이브 밑에 mytext23.txt로 저장하시오!
(반드시 저장해서 기사를 워드클라우드로 시각화를 할 수 있으니까 오늘 배운 내용으로 저장하세요)

```
from bs4 import BeautifulSoup
f = open("c:₩₩data₩₩aa77.html", encoding='UTF8')
soup = BeautifulSoup( f, "html.parser")

result = soup.find_all(class_ = 'article_body mg fs4')


a = []
for i in result:
a.append(i.get_text())
```

```
jj= open("c:₩₩data₩₩mytext23.txt", "w")
jj.write(str(a))
jj.close()
```

415. (점심시간 문제) 레이디버그 게시판의 전체 게시글은 총 몇건인가?

코드와 답을 올리세요~

```
# 0. 웹스크롤링에 필요한 모듈을 임포트 합니다.
from bs4 import BeautifulSoup
import urllib.request #웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 1. 레이디 버그 게시판의 게시글을 모두 params에 담습니다.
params=[]
for i in range(1,23):
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+ str(i) + '&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")
    result2 = soup.find_all( 'p', class_ ='con')
    for i in result2:
        params.append(i.get_text(" ", strip=True))
#2. 게시글의 건수를 확인합니다.
print(len(params))
```

416. 게시글 429개 전체를 c드라이브 밑에 mytext34.txt라는 이름으로 저장하시오.

- 텍스트 파일을 저장하는 파이썬 코드

```
text = 'abcdefghijklmn'
f = open('c:₩₩data₩₩mytext32.txt', 'w', encoding='UTF8')
f.write(text)
f.close()
```

```
# 1. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱
from  bs4  import  BeautifulSoup
import  urllib.request   # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

params1 = []
params2 = []

f = open('c:₩₩data₩₩mytext34.txt', 'w', encoding='UTF8')

for  i  in  range(1, 23):
    list_url = 'http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' +str(i) + '&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
```

```
    url = urllib.request.Request(list_url) #  사람이 알아볼 수 있는 위의 url 을 파이썬이 알아
                                                # 볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")

    # 2. 시청자 게시판의 날짜와 본문 내용을 가져옵니다.
    result1 = soup.find_all( 'span', class_ ='date')
    result2 = soup.find_all( 'p',  class_ ='con')

    # 3. 시청자 게시판의 날짜와 본문을 params 와 params2 리스트에 담습니다.

    for  i  in  result1:
        params1.append( i.get_text("  ", strip=True) )
    for  i  in  result2:
        params2.append( i.get_text("  ", strip=True) )

# 4. 날짜와 본문을 같이 출력 합니다.
for   k,  h   in   zip(  params1,  params2 ):
    f.write ( k + '   ' + h + '₩n' )

f.close()
```

417. 동아일보에서 검색 키워드(예: 인공지능)를 가지고 검색을 한 후 그 url을 얻어낸다.

https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1

https://www.donga.com/news/search?p=16&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1

https://www.donga.com/news/search?p=31&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1

418. 상세기사 url을 리스트에 담는 d_scroll() 함수를 생성하시오.

p테그 클레스 txt

```
def d_scroll():
    list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")
```

```
    result1 = soup.find_all( 'p', class_ = 'txt')

    params = []

    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
            params.append(k.get('href'))

    return params

print(d_scroll())
```

419. 상세기사 url로 기사본문을 스크롤링하는 d_detail_scroll() 함수를 생성한다.
div 테그, "article txt"

답:

```
def d_scroll():
    list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    result1 = soup.find_all( 'p', class_ = 'txt')

    params = []

    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
            params.append(k.get('href'))

    return params

def d_detail_scroll():
    list_url = d_scroll()
    params2 = []
    for i in list_url:
        url = urllib.request.Request(i) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
        result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
        soup = BeautifulSoup( result, "html.parser")

        result1 = soup.find_all( 'div', class_ = 'article_txt')


        for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )
```

```
    return params2

print(d_detail_scroll())
```

420. 한겨레 신문사에서 인공지능으로 검색했을때 나오는 기사 본문을 스크롤링하는 함수 두개를 생성하시오.

h_scroll() : 상세기사 url을 가져오는 함수

h_detail_scroll() : 상세기사 url로기사 본문을 가져오는 함수

url:
http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=0

http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=1

답:

```
def h_scroll():
    list_url = 'http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=0'
    url = urllib.request.Request(list_url) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
    result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의 url의 html문서들을 result변수에 담음.
    soup = BeautifulSoup( result, "html.parser")

    result1 = soup.find_all( 'dt')

    params = []

    for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
        for k in i:   #h2 테그안의 a테그의 html코드를 가져오기 위한 코드
            params.append('http:' + k.get('href'))

    return params

def h_detail_scroll():
    list_url = h_scroll()
    params2 = []
    for i in list_url:
        url = urllib.request.Request(i) #사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환 첫번째 작업
        result = urllib.request.urlopen(url).read().decode("UTF-8") #두번째 작업, 위의
```

```
url의 html문서들을 result변수에 담음.
        soup = BeautifulSoup( result, "html.parser")

        result1 = soup.find_all( 'div', class_ = 'text')


        for i in result1: #result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )

    return params2

print(h_detail_scroll())
```

421. (오늘의 마지막 문제 12/16) 원하는 신문사에서 스크롤링 하고싶은 키워드를 넣고 검색했을 때 나오는 본문기사들을 수집하는 함수 2개를 생성하시오~ (밑에 페이지번호 1,2,3만 스크롤링하세요~)

1. JJ_scroll() : 상세기사 url 가져오는 함수
2. jj_detail_scroll() : 상세기사 url

https://search.etnews.com/etnews/search.php?category=CATEGORY1&kwd=%EC%BD%94%EB%A1%9C%EB%82%98&pageNum=1&pageSize=3&reSrchFlag=false&sort=1&startDate=&endDate=&sitegubun=&jisikgubun=&preKwd%5B0%5D=%EC%BD%94%EB%A1%9C%EB%82%98

https://search.etnews.com/etnews/search.php?category=CATEGORY1&kwd=%EC%BD%94%EB%A1%9C%EB%82%98&pageNum=2&pageSize=10&reSrchFlag=false&sort=1&startDate=&endDate=&sitegubun=&jisikgubun=&preKwd%5B0%5D=%EC%BD%94%EB%A1%9C%EB%82%98

----------실패----------

422. (오늘의 마지막 문제12/17) 머신러닝 때 나이브 베이즈 확률을 배울 때 사용하기 위해서 네이버 영화 평점 게시판의 게시글들을 스크롤링 해서 영화 이름을 텍스트 파일로 해서 저장히시오~

코드와 텍스트 파일을 첨부해서 올리세요~

영화는 자유롭게 선택하세요!

비긴어게인
https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=96379&target=after&page=1

답:

```
from bs4 import BeautifulSoup
import urllib.request
```

```
params1 = []
params2 = []

mgrade= open("c:₩₩data₩₩navermovie.txt", "w", encoding='UTF8') #평점댓글을 저장할 수 있는 파일을 만듦
for i in range(1, 1001): #네이버 평점은 1000페이지까지 볼 수 있다고 함
    list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=96379&target=after&page=' + str(i)
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read()
    soup = BeautifulSoup( result, "html.parser")
    result1 = soup.find_all('td', class_ = 'title') # 평점관련 댓글
    result2 = soup.find_all('td', class_ = 'num')   # 댓글을 달은 날짜

    for i in result1:
        params1.append(i.get_text(" ", strip=True))
    for i in result2:
        params2.append(i.get_text(" ", strip=True))

for k, h in zip(params1, params2):
    mgrade.write(k+' ' + h + "₩n")

mgrade.close()
```

423. 어제 조선일보에서 봉사로 검색햇을때 다운 받았던 기사에서 가장 많이 나오는 단어(어절)가 무엇인가?

```
stev = open("c:₩₩data₩₩bongsa.txt", encoding="UTF8")
stev2 = stev.read().split()
stev2.sort()
print(stev2)
```

결과:

['&', '&', '&', '&', '&', '&', '&', '&', '&', '&', '&', '&', '(예비)사회적기업,', '10기)]–', '10대', '10분', '10시쯤', '10월', '11가지', '13292)에', '1365자원봉사포털(www.1365.go.kr)에서', '13일,', '13일까지', '1457명이', '1500개가량의', '157', '15명(선착순)이며,', '15일(현지', '16일(이하', '1785명의', '17개의', '17일', '1800개', '1800개', '1800개.', '18일(금)까지', '1961년', '1990년부터', '1991년부터', '19:00부터', '19세', '19일까지이며,', '1년', '1년간이며,', '1년을', '1분이면', '1시간', '1시부터', '1인당', '1층', '2002년', '2006년', '2009년', '2009년부터는', '200개를', '2017년', '2018년', '2019년', '2021년까지', '2089명이', '20·30대', '20개다.', '20대다.', '22일', '23일', '24일', '24일부터', '26명이', '27일', '27일까지', '28)씨를', '28일', '28일에도', '29일부터', '2년', '2년', '2년까지', '2년을', '2명을', '2시간', '2월', '2인', '2일', '300여명이', '30만원씩', '30수', '30여', '30일', '31개국에', '31명의', '31일까지', '31일까지', '31일까지', '33명이', '36.2도.', '3년간', '3년에', '3년이', '3단계(▲서류', '3시간', '3시간', '3시간)', '3시부터', '3월', '3일', '3일(목)', '3회(1회', '400명이', '40명을', '42국에서', '4500명의', '4개', '4개이며,', '4박', '4월', '4일', '5000만', '54개국에서', '5곳을', '5만명에', '5명과', '5시간을', '5시까지', '5시까지,', '5시부터', '5시에', '5시에', '5시에', '5월', '5월', '5월', '5일간', '61국에서', '6~8개팀,', '6개월', '6박', '6일까지다.', '71국에서', '7300여', '7명을', '7층,', '80개의', '8월', '8일이며,', '96만', '9시', 'A씨', 'A씨가', 'A씨가', 'A씨는', 'A씨는', 'A씨는', 'A씨에게', 'A씨의', 'CNN', 'Cares)’를', 'Cares)’에는',

'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Copyrights', 'Corp)의', 'D.I.Y', 'D.I.Y', 'Filter)'', 'Friends', 'Friends', 'KCOC', 'KCOC', 'KCOC(국제개발협력민간협의회)에', 'KCOC가', 'KOICA', 'KOICA', 'KOICA', 'Korea·이하', 'Korea·이하', 'LAB2050', 'MG사회적경제기업', 'MG새마을금고중앙회X함께일하는재단,', 'NGO', 'NGO', 'NGO', 'NGO', 'NGO', 'NGO', 'NGO', 'NGO', 'NGO,', 'NGO가', 'NGO가', 'NGO가', 'NGO들의', 'NGO들의', 'NGO를', 'NGO를', 'NGO를', 'NGO에', 'NGO에', 'NGO에', 'NGO에', 'NPO', 'Permit)을', 'SRGs는', 'TV를', 'UN', 'WFK', 'WFK', 'WFK', 'WFK)', 'WFK)'', 'WFK-NGO', 'WFK-NGO', 'WFK-NGO', 'WFK-NGO', 'WFK는', 'WFK는', 'WFK라는', 'WFK소속', 'York', '[더', '[문일요', '[박민영', '[박선하', '[박혜연', '[어떻게', '[우리사회', '[이영재', 'bada@chosun.com]', 'bbb', 'bbb', 'bbb', 'bbb', 'bbb', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'futurechosun.com,', 'honey@chosun.com]–', 'ilyo@chosun.com]–', 'it', 'sona@chosun.com', 'sona@chosun.com]', 'yourself)'', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', '–', ''1차', ''2016~2018', ''2018', ''2차', ''D.I.Y(Do', ''KF(Korea', ''MG사회적경제기업', ''NGO가', ''NGO비자'나', ''SRGs(사회적', ''WFK-NGO', ''WFK'', ''공공', ''관광·학생', ''관성탈출'', ''관성탈출'', ''그간', ''글로벌', ''깐깐' ....]

```
stev = open("c:₩₩data₩₩bongsa.txt", encoding="UTF8")
stev2 = stev.read().split() #어절별로 분리해서 리스트에 담는다.
stev2.sort() #리스트안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) #stev2리스트 안의 어절별로 각각 몇건씩 있는지 정리를 합니다.
result = count_list.most_common(10) #TOP10개만 추출
print(result)
```

결과:

[('수', 33), ('있다.', 28), ('했다.', 24), ('등', 22), ('–', 19), ('더나은미래', 17), ('비자', 17), ('봉사', 16), ('마스크', 15), ('있는', 15)]

424. 문제 423번 결과를 for loop문을 이용해서 아래와 같이 출력하시오!

수 33

있다 28

했다 24

: :

답:

```
stev = open("c:₩₩data₩₩bongsa.txt", encoding="UTF8")
stev2 = stev.read().split() #어절별로 분리해서 리스트에 담는다.
stev2.sort() #리스트안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) #stev2리스트 안의 어절별로 각각 몇건씩 있는지 정리를 합니다.
result = count_list.most_common(30) #TOP10개만 추출
for i in result:
```

```
    print(i[0],i[1])
```

425. 어제 마지막 문제로 스크롤링 했던 영화 평가 게시글들에서 위와 같이 가장 많이 나오는 어절이 무엇인지 출력하시오.

```
stev = open("c:₩₩data₩₩navermovie.txt", encoding="UTF8")
stev2 = stev.read().split() #어절별로 분리해서 리스트에 담는다.
stev2.sort() #리스트안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) #stev2리스트 안의 어절별로 각각 몇건씩 있는지 정리를 합니다.
result = count_list.most_common(30) #TOP10개만 추출
for i in result:
    print(i[0],i[1])
```

```
중 10033
비긴 10022
10점 10013
- 10004
어게인 9996
별점 9985
총 9982
신고 9980
10 5967
9 1534
영화 1291
8 1151
너무 1005
정말 748
7 489
좋은 428
노래가 413
음악이 407
진짜 406
영화를 344
영화. 336
노래 329
더 329
좋고 321
6 302
이 289
최고의 285
그냥 281
보고 273
노래도 267
```

426. 웹스크롤링을 한 데이터로 감정분석(긍정적, 부정적)을 하기 위해 신성이의 라라랜드 영화 평과를 다운로드 해서 c 밑에 data밑에 두고 가장 많이 나오는 단어가 무엇인지 확인합니다.

답:

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding="UTF8")
stev2 = stev.read().split() #어절별로 분리해서 리스트에 담는다.
stev2.sort() #리스트안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) #stev2리스트 안의 어절별로 각각 몇건씩 있는지 정리를 합니다.
result = count_list.most_common(30) #TOP10개만 추출
for i in result:
    print(i[0],i[1])
```

결과:

```
10점 100
중 100
10 79
영화 22
이 12
다시 10
수 9
본 7
9 6
그 6
그때 6
내 6
너무 6
보고 6
봐도 6
8 5
내가 5
더 5
번 5
않았다면.만약에 5
있는 5
잘 5
한 5
것 4
다 4
마지막 4
만약에, 4
아주 4
영화가 4
왜 4
```

427. 영화 라라랜드에서 긍정적인 단어가 몇건이 나오는지 출력하시오!
(문제 276번 코드를 가져오시오)

```
positive = open("c:₩₩data₩₩positive-words2.txt", encoding = 'CP949')
pos = positive.read().split('₩n')

stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.read().split()
```

```
cnt = 0
for i in stev2:          #stev3리스트에서 하나씩 빼냅니다.
      if i.lower() in pos:          #i의 단어가 pos긍정단어 리스트에 존재하면
        cnt = cnt +1  #cnt를 1씩 증가시킵니다.
print(cnt) #37
```

428. 라라랜드 영화 평가에 부정단어는 몇개가 있고 단어들은 무엇인지 출력하시오!

```
negative = open("c:₩₩data₩₩negative-words2.txt", encoding = 'CP949')
pos = negative.read().split('₩n')

stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.read().split()

cnt = 0
for i in stev2:          #stev3리스트에서 하나씩 빼냅니다.
      if i.lower() in pos:          #i의 단어가 pos긍정단어 리스트에 존재하면
        cnt = cnt +1  #cnt를 1씩 증가시킵니다.
print(cnt) #17
```

429. 영화 라라랜드의 평가글들을 워드 클라우드로 시각화 하시오!

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt  # 그래프 그리는 모듈
from os import path     #  os 에 있는 파일을 파이썬에서 인식하기 위해서
import re   #  데이터 정제를 위해서 필요한 모듈
import numpy as np
from PIL import Image  # 이미지 시각화를 위한 모듈

# 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:₩₩project₩₩usa_im.png"))

# 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
script = 'sample6_laland_review.txt'

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
d = path.dirname("c:₩₩project₩₩")

# 기사 스크립트와 os 의 위치를 연결하여 utf8로 인코딩해서 한글 텍스트를
# text 변수로 리턴한다.
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()
print(text)

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
file = open('c:/project/word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') #어절별로 분리해서 word에 담아 리스트로 구성한다.
for i in word:
    text = re.sub(i,'',text)  #re.sub('있다', '', '있다') <--라라랜드 게시글에 있다를 ''으로
대체하겠다.
#설명: 일반적인 문장에서 자주나오는 단어들을 전부 ''로 대체하겠다.
#re.sub('있다','','있다')
#re.sub('하지만','','하지만')
```

```
# 워드 클라우드를 그린다.
wordcloud = WordCloud(font_path='C://Windows//Fonts//gulim', # 글씨체
                      stopwords=STOPWORDS,   # 마침표, 느낌표,싱글 쿼테이션 등을 정
제
                      max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
                      background_color='white', # 배경색깔
                      max_font_size = 100, # 최대 글씨 크기
                      min_font_size = 1, # 최소 글씨
                      mask = usa_mask, # 배경 모양
                      colormap='jet').generate(text).to_file('c:/project/lala_cloud.png')
                  # c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이
름

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')  # 글씨가 퍼지는 스타일
plt.axis("off") #축 표시 없음
```

430. (점심시간 문제) 어제 마지막 문제로 받았던 영화 평가들 또는 어제 스크롤링했던 기사들을 하나 선택해서 워드 클라우드로 그리고 그림을 올리고 검사받으세요~~

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt  # 그래프 그리는 모듈
from os import path     #  os 에 있는 파일을 파이썬에서 인식하기 위해서
import re   #  데이터 정제를 위해서 필요한 모듈
import numpy as np
from PIL import Image  # 이미지 시각화를 위한 모듈

# 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:₩₩project₩₩usa_im.png"))

# 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
script = 'navermovie.txt'

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
d = path.dirname("c:₩₩project₩₩")

# 기사 스크립트와 os 의 위치를 연결하여 utf8로 인코딩해서 한글 텍스트를
# text 변수로 리턴한다.
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
file = open('c:/project/word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') #어절별로 분리해서 word에 담아 리스트로 구성한다.
for i in word:
    text = re.sub('신고','',text)
    text = re.sub('별점', '',text)
    text = re.sub('10점', '', text)
    text = re.sub('중', '', text)
    text = re.sub('총', '', text)#re.sub('있다', '', '있다') <--라라랜드 게시글에 있다를 ''으
로 대체하겠다.
#설명: 일반적인 문장에서 자주나오는 단어들을 전부 ''로 대체하겠다.
#re.sub('있다','','있다')
```

```
#re.sub('하지만','','하지만')

# 워드 클라우드를 그린다.
wordcloud = WordCloud(font_path='C://Windows//Fonts//gulim', # 글씨체
                      stopwords=STOPWORDS,   # 마침표, 느낌표,싱글 쿼테이션 등을 정
제
                      max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
                      background_color='white', # 배경색깔
                      max_font_size = 100, # 최대 글씨 크기
                      min_font_size = 1, # 최소 글씨
                      mask = usa_mask, # 배경 모양

colormap='jet').generate(text).to_file('c:/project/beginagain_cloud.png')
                  # c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이
름

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')  # 글씨가 퍼지는 스타일
plt.axis("off") #축 표시 없음
```

431. 신성이가 받은 라라랜드 평가 게시글에서 평가 점수만 출력하시오!

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()
for i in stev2:
    print(i[6:8])  #type: <class 'str'>
```

그래서 숫자로 바꿔준다.

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()
for i in stev2:
    print( int( i[6:8] ) )
```

432. 라라랜드 평가글중에 평점이 6점이상인 글들만 출력하시오!

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()
for i in stev2:
    if int(i[6:8]) >=6:
        print(i, end="") #end쓰면 예쁘게 나온다.
```

433. 라라랜드 평가글중에 평점이 6점 이상이면 pos라는 비어있는 리스트에 해당 평가글을 append되게 하고 6점 미만이면 nag라는 비어있는 리스트에 평가글이 append 되게 하시오!

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()
pos = []
nag = []
for i in stev2:
```

```
    if int(i[6:8]) >=6:
        pos.append(i[8:]) #10점 중 ~ 가 안나오게 담기
    else:
        nag.append(i[8:])

print(pos)
print(nag)
```

434. 위의 pos에 들어있는 긍정글들을 c드라이브 밑에 project 밑에 pos_lala.txt로 저장하시오!

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()

pos = []
nag = []
for i in stev2:
    if int(i[6:8]) >=6:
        pos.append(i[8:])
    else:
        nag.append(i[8:])


f = open("c:₩₩project₩₩pos_lala.txt", "w")
f.writelines(pos)
f.close()
```

435. 6점 미만인 글들은 nag_lala.txt로 c드라이브 밑에 project밑에 저장되게 하시오!

```
stev = open("c:₩₩data₩₩sample6_laland_review.txt", encoding = 'UTF8')
stev2 = stev.readlines()

pos = []
nag = []
for i in stev2:
    if int(i[6:8]) >=6:
        pos.append(i[8:])
    else:
        nag.append(i[8:])


f = open("c:₩₩project₩₩pos_lala.txt", "w")
f.writelines(pos)
f.close()

f2 = open("c:₩₩project₩₩nag_lala.txt", "w")
f2.writelines(nag)
f2.close()
```

436. 그러면 위의 pos_lala.txt의 글과 nag_lala.txt의 글을 워드 클라우드로 각각 시각화하시오!

(파일이 있어야 가능함)

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt  # 그래프 그리는 모듈
from os import path     #  os 에 있는 파일을 파이썬에서 인식하기 위해서
import re   #  데이터 정제를 위해서 필요한 모듈
import numpy as np
from PIL import Image  # 이미지 시각화를 위한 모듈

# 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:₩₩project₩₩usa_im.png"))

# 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
script = 'pos_lala.txt'

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
d = path.dirname("c:₩₩project₩₩")

# 기사 스크립트와 os 의 위치를 연결하여 utf8로 인코딩해서 한글 텍스트를
# text 변수로 리턴한다.
text = open(path.join(d, "%s"%script), mode="r", encoding="cp949").read()

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
file = open('c:/project/word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') #어절별로 분리해서 word에 담아 리스트로 구성한다.
for i in word:
    text = re.sub(i,'',text)  #re.sub('있다', '', '있다') <--라라랜드 게시글에 있다를 ''으로
대체하겠다.
#설명: 일반적인 문장에서 자주나오는 단어들을 전부 ''로 대체하겠다.
#re.sub('있다','','있다')
#re.sub('하지만','','하지만')

# 워드 클라우드를 그린다.
wordcloud = WordCloud(font_path='C://Windows//Fonts//gulim', # 글씨체
                      stopwords=STOPWORDS,   # 마침표, 느낌표,싱글 쿼테이션 등을 정
제
                      max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
                      background_color='white', # 배경색깔
                      max_font_size = 100, # 최대 글씨 크기
                      min_font_size = 1, # 최소 글씨
                      mask = usa_mask, # 배경 모양
                      colormap='jet').generate(text).to_file('c:/project/lala_cloud.png')
                  # c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이
름

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')  # 글씨가 퍼지는 스타일
plt.axis("off") #축 표시 없음
```

437. 평점 좋은 영화 데이터만 뽑아보기

438. c드라이브 밑에 gimages2 라는 폴더를 만들고 거기에 다른 이미지를 다운로드 받으세요~

```
##########################################################
import urllib.request # 웹 url을 파이썬이 인식 할 수 있게하는 패키지
from  bs4 import BeautifulSoup # html에서 데이터 검색을 용이하게 하는 패키지
from selenium import webdriver
# selenium : 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크
# 손으로 클릭하는 것을 컴퓨터가 대신하면서 스크롤링하게 하는 패키지

from selenium.webdriver.common.keys import Keys
import time        # 중간중간 sleep 을 걸어야 해서 time 모듈 import

############################ url 받아오기
############################

# 웹브라우져로 크롬을 사용할거라서 크롬 드라이버를 다운받아 아래 파일경로의 위치에 둔다
# 팬텀 js로 하면 백그라운드로 실행할 수 있음
binary = 'c:₩chromedriver/chromedriver.exe'

# 브라우져를 인스턴스화
browser = webdriver.Chrome(binary)

# 구글의 이미지 검색 url 받아옴(아무것도 안 쳤을때의 url)
browser.get("https://www.google.co.kr/imghp?
hl=ko&tab=wi&ei=l1AdWbegOcra8QXvtr-4Cw&ved=0EKouCBUoAQ")

# 구글의 이미지 검색에 해당하는 input 창의 id 가 '  ?  ' 임(검색창에 해당하는 html 코드를 찾아서 elem 사용하도록 설정) ?: id가 뭔지 알아낼 것
# input창 찾는 방법은 원노트에 있음

# elem = browser.find_elements_by_class_name('gLFyf gsfi') # Tip : f12누른후 커서를 검색창에 올려두고 id를 찾으면 best

elem = browser.find_element_by_xpath("//*[@class='gLFyf gsfi']")  # 위의 코드대로 하거나 이렇게 하거나 둘 중 하나 select


############################ 검색어 입력
############################

# elem 이 input 창과 연결되어 스스로 햄버거를 검색
elem.send_keys("수지") # 여기에 스크롤링하고싶은 검색어를 입력

# 웹에서의 submit 은 엔터의 역할을 함
elem.submit()

# 현재 결과 더보기는 구현 되어있지 않은상태 -> 구글의 경우 400개 image가 저장됨.

############################ 반복할 횟수
############################

# 스크롤을 내리려면 브라우져 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번 하고 End키를 눌러야함
```

```
for i in range(1, 6): # 5번 스크롤 내려가게 구현된 상태 range(1,5)
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(10)          # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 쉼

time.sleep(10)                      # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)
html = browser.page_source          # 크롬브라우져에서 현재 불러온 소스 가져옴
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


browser.find_element_by_xpath("//*[@class='mye4qd']").click()  # 여기가 결과 더보기 코드입니다

for i in range(1, 5): # 4번 스크롤 내려가게 구현된 상태 range(1,5)
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(10)          # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 쉼

time.sleep(10)                      # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)
html = browser.page_source          # 크롬브라우져에서 현재 불러온 소스 가져옴
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정

############################ 그림파일 저장
############################

### 검색한 구글 이미지의 url을 따오는 코드 ###
def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="rg_i Q4LuWd")  # 구글 이미지 url 이 있는 img 태그의 _img 클래스에 가서 (f12로 확인가능.)
    for im in imgList:
        try :
            params.append(im["src"])                    # params 리스트에 image url 을 담음.
        except KeyError:
            params.append(im["data-src"])
    return params

# except부분
# 이미지의 상세 url 의 값이 있는 src 가 없을 경우
# data-src 로 가져오시오 ~


def fetch_detail_url():
    params = fetch_list_url()

    for idx,p in enumerate(params,1):
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "C:/gimages2/" + str(idx) + "_google.jpg")
```

```
# enumerate 는 리스트의 모든 요소를 인덱스와 쌍으로 추출
# 하는 함수 . 숫자 1은 인덱스를 1부터 시작해라 ~

fetch_detail_url()

# 끝나면 브라우져 닫기
browser.quit()
```

439. (오늘의 마지막 문제 12/18) 마이크로소프트 회사에서 만든 검색페이지인 bing에서 이미지 검색을 할 수 있도록 하는 웹스크롤링 코드를 작성하시오!
(네이버 코드를 좀 수정하면 됩니다.)

딥러닝 수업 전까지 학습하고 싶은 이미지 두가지를 정해서 사진을 스크롤링 해놓기!!
(한클래스 당 2000장)

```
import  urllib.request   #웹url을 파이썬이 인식할 수 있게 하는 패키지
from  bs4  import  BeautifulSoup #html 코드에서 원하는 지점을 빨리 찾을 수 있게 만든 모듈
from selenium import webdriver  #손으로 클릭하는것을 컴퓨터가 하게끔 하는 모듈
from selenium.webdriver.common.keys import Keys
import time #중간중간 sleep을 주려고 임포트 함

binary = 'C:₩chromedriver/chromedriver.exe' #크롬 드라이버 위치 지정
browser = webdriver.Chrome(binary) #browser 객체 생성
browser.get("https://www.bing.com/images?FORM=Z9LH")
elem = browser.find_element_by_id("sb_form_q") # 검색창 지점을 알아내서 elem에 담는다.
#find_elements_by_class_name("") #클래스 이름으로 찾을 때 필요한 코드

# 검색어 입력
elem.send_keys("디즈니")  # 컴퓨터가 디즈니 글씨를 직접 적는다.
elem.submit()                  # 그리고 엔터를 친다.

# 반복할 횟수
for i in range(1,5): #end키를 누르면서 아래로 내리는데
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(10)   #슬립을 10초 주면서 5번 수행한다.

time.sleep(10) # 10초 멈췄다가

html = browser.page_source  # 현 페이지의 html 코드를 불러와서
soup = BeautifulSoup(html,"lxml")  # BeautifulSoup을 이용할 수 있도록 파싱한다.


def fetch_list_url():  #이미지의 상세 url 가져오는 함수
    params = []
    imgList = soup.find_all("img", class_="mimg") #img 테그의 클래스 이름 mimg로 접근
    for im in imgList:
        params.append(im["src"])  #src의 값을 가져와서 params에 append 합니다.
    return params
```

```
def  fetch_detail_url(): #상세 이미지 url 가져와서 이미지 다운로드하는 함수
    params = fetch_list_url()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/bing/"+ str(a) + ".jpg" )

        a = a + 1

fetch_detail_url()

browser.quit()
```

440. 오라클의 emp테이블 전체를 select했는데 전체를 다 select하지 말고 아래의 쿼리의 결과만 select하시오!

```
select empno, ename, sal, deptno
      from emp
```

답:

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유져 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select empno, ename, sal, deptno from emp""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

441. dept테이블의 모든 데이터를 조회하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유져 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select * from dept""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
dept = pd.DataFrame(row)
print(dept)
```

442. 우리반테이블(emp12)를 조회하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select * from emp12""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp12 = pd.DataFrame(row)
print(emp12)
```

443. 월급이 1200 이상인 사원들의 이름과 월급을 출력하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select ename, sal from emp where sal>=1200""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

444. 사원 테이블 월급을 막대그래프로 그리시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select ename, sal from emp""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
emp.index = list(emp[0])
emp.plot(kind='bar')
```

445. (점심시간 문제) 문제 444번의 그래프의 색깔을 변경하고 그래프를 올리세요.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
```

```
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select ename, sal from emp""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
emp.index = list(emp[0])
emp.plot(kind='bar', color = 'orange')
```

446. 직업, 직업별 토탈 월급을 구하시오

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select job, sum(sal) from emp group by job""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.

row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

447. 직업, 직업별 토탈월급을 출력하는데 직업별 토탈월급이 높은것부터 출력하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
```

```
cursor.execute("""select job, max(sal) from emp group by job order by max(sal)
desc""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.

row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

448. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 사원 순으로 순위를 부여하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select ename, sal, dense_rank() over (order by sal desc)
            from emp""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

449. 부서번호, 부서번호별 토탈월급을 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select deptno, sum(sal)
            from emp
            group by deptno""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
print(emp)
```

450. 문제 449번의 결과를 막대그래프로 시각화 하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl') #오라클 주소를 기입한다.
db = cx_Oracle.connect('scott', 'tiger', dsn) #오라클 접속 유저 정보
cursor = db.cursor() # 결과 데이터를 담을 메모리 이름을 cursor로 선언함
cursor.execute("""select deptno, sum(sal)
            from emp
            group by deptno""") #작성한 쿼리문의 결과가 cursor메모리에 담긴다.
row = cursor.fetchall() #cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
emp = pd.DataFrame(row)
```

```
emp.index = list(emp[0])
emp.plot(kind = 'bar', color = 'red')
```

451. emp 테이블 전체를 출력하는데 컬럼명이 붙어서 출력되게하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
                from emp""")
row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description  #컬럼명을 가져온다.
print(colname)
```

결과:

```
[('EMPNO', <cx_Oracle.DbType DB_TYPE_NUMBER>, 5, None, 4, 0, 0), ('ENAME',
<cx_Oracle.DbType DB_TYPE_VARCHAR>, 10, 10, None, None, 1), ('JOB',
<cx_Oracle.DbType DB_TYPE_VARCHAR>, 9, 9, None, None, 1), ('MGR',
<cx_Oracle.DbType DB_TYPE_NUMBER>, 5, None, 4, 0, 1), ('HIREDATE',
<cx_Oracle.DbType DB_TYPE_DATE>, 23, None, None, None, 1), ('SAL',
<cx_Oracle.DbType DB_TYPE_NUMBER>, 11, None, 7, 2, 1), ('COMM',
<cx_Oracle.DbType DB_TYPE_NUMBER>, 11, None, 7, 2, 1), ('DEPTNO',
<cx_Oracle.DbType DB_TYPE_NUMBER>, 3, None, 2, 0, 1)]
```

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
                from emp""")
row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
print(col)
```

결과:

```
['empno', 'ename', 'job', 'mgr', 'hiredate', 'sal', 'comm', 'deptno']
```

452. 문제 451의 col 리스트에담긴 컬럼명을 위의 데이터에 매핑시키시오.

```
import cx_Oracle
```

```
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
                from emp""")
row = cursor.fetchall()
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
emp = pd.DataFrame(list(row), columns=col)
print(emp)
```

453. 문제 451의 컬럼명을이용 해서 판다스 문법으로이름과 월급을 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
                from emp""")
row = cursor.fetchall()
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
emp = pd.DataFrame(list(row), columns=col)
print(emp[['ename', 'sal']])
```

454. 월급이 3000 이상인 사원들의 이름과 월급을 출력하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
                from emp""")
row = cursor.fetchall()
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
emp = pd.DataFrame(list(row), columns=col)
print(emp[['ename', 'sal']][emp['sal']>=3000])
```

455. 이름과 부서위치를 출력하시오! (SQL로 작성해서 파이썬으로 출력되게 하시오.)

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select e.ename, d.loc
                  from emp e, dept d
                  where e.deptno = d.deptno""")
row = cursor.fetchall()
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
emp = pd.DataFrame(list(row), columns=col)
print(emp)
```

456. 부서위치, 부서위치 토탈월급을 출력하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
                  from emp e, dept d
                  where e.deptno = d.deptno
                  group by d.loc
                  """)
row = cursor.fetchall()
colname = cursor.description

col = []
for i in colname:
    col.append(i[0].lower())
emp = pd.DataFrame(list(row), columns=col)
print(emp)
```

457. 문제 456 결과를 막대그래프로 시각화 하시오!

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
                  from emp e, dept d
                  where e.deptno = d.deptno
```

```
                    group by d.loc
                    """)
row = cursor.fetchall()

emp = pd.DataFrame(row)
emp.index = list(emp[0])
emp.plot(kind = 'bar', color = 'red')
```

458. 월급이 3000이상인 사원들의 이름과 월급을 출력하시오!

```
mysql> select ename, sal
    -> from emp
    -> where sal >= 3000;
```

결과:

```
+-------+------+
| ename | sal  |
+-------+------+
| KING  | 5000 |
| FORD  | 3000 |
| SCOTT | 3000 |
+-------+------+
```

459. 직업 SALESMAN인 사원들의 이름 월급과 직업을 출력하는데 월급이 높은사원들부터 출력하시오!

```
mysql> select ename, sal, job
    -> from emp
    -> where job = 'SALESMAN'
    -> order by sal desc;
```

결과:

```
+--------+------+----------+
| ename  | sal  | job      |
+--------+------+----------+
```

```
| TURNER | 1500 | SALESMAN |
| MARTIN | 1250 | SALESMAN |
| WARD   | 1250 | SALESMAN |
+--------+------+----------+
```

460. 이름과 부서위치를 출력하시오.

```
mysql> select e.ename, d.loc
    -> from emp e, dept d
    -> where e.deptno = d.deptno;
```

결과:

```
+--------+----------+
| ename  | loc      |
+--------+----------+
| KING   | NEW YORK |
| BLAKE  | CHICAGO  |
| CLARK  | NEW YORK |
| JONES  | DALLAS   |
| MARTIN | CHICAGO  |
| TURNER | CHICAGO  |
| JAMES  | CHICAGO  |
| WARD   | CHICAGO  |
| FORD   | DALLAS   |
| SMITH  | DALLAS   |
| SCOTT  | DALLAS   |
| ADAMS  | DALLAS   |
| MILLER | NEW YORK |
+--------+----------+
```

461. JONES 보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오! (서브쿼리문)

```
mysql> select ename, sal
    -> from emp
    -> where sal > (select sal
    ->                 from emp
    ->                 where ename = 'JONES');
```

결과:

```
+-------+------+
| ename | sal  |
+-------+------+
| KING  | 5000 |
| FORD  | 3000 |
| SCOTT | 3000 |
+-------+------+
```

462. 이름, 월급, 순위를 출력하시오!

(순위는 월급 높은 순으로 순위를 부여하세요!)

```
mysql> select ename, sal, dense_rank() over(order by sal desc) 순위
    -> from emp;
```

결과:

```
+--------+------+------+
| ename  | sal  | 순위 |
+--------+------+------+
| KING   | 5000 |    1 |
| FORD   | 3000 |    2 |
| SCOTT  | 3000 |    2 |
| JONES  | 2975 |    3 |
| BLAKE  | 2850 |    4 |
| CLARK  | 2450 |    5 |
| TURNER | 1500 |    6 |
| MILLER | 1300 |    7 |
| MARTIN | 1250 |    8 |
| WARD   | 1250 |    8 |
| ADAMS  | 1100 |    9 |
| JAMES  |  950 |   10 |
| SMITH  |  800 |   11 |
+--------+------+------+
```

463. 이름, 커미션을 출력하는데 커미션이 null인 사원들은 0으로 출력하시오!

```
mysql> select ename, ifnull(comm, 0)
    -> from emp;
```

결과:

```
+--------+-----------------+
| ename  | ifnull(comm, 0) |
+--------+-----------------+
| KING   |               0 |
| BLAKE  |               0 |
| CLARK  |               0 |
| JONES  |               0 |
| MARTIN |            1400 |
| TURNER |               0 |
| JAMES  |               0 |
| WARD   |             500 |
| FORD   |               0 |
| SMITH  |               0 |
| SCOTT  |               0 |
| ADAMS  |               0 |
| MILLER |               0 |
+--------+-----------------+
```

464. 오늘 날짜를 출력하시오!

```
mysql> select sysdate()
    -> ;
```

결과:

```
+---------------------+
| sysdate()           |
```

```
+---------------------+
| 2020-12-21 15:12:05 |
+---------------------+
```

설명 : mySQL은 오라클과 같은 dual이 없음

465. 부서번호, 부서번호별 토탈월급을 출력하시오!

```
mysql> select deptno, sum(sal)
    -> from emp
    -> group by deptno;
```

결과:

```
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|     10 |     8750 |
|     30 |     7800 |
|     20 |    10875 |
+--------+----------+
```

466. 문제 465번 결과를 다시 출력하는데 맨 아래 전체 토탈월급이 출력되게 하시오!

```
mysql> select deptno, sum(sal)
    -> from emp
    -> group by deptno with rollup;
```

결과:

```
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|     10 |     8750 |
|     20 |    10875 |
|     30 |     7800 |
|   NULL |    27425 |
+--------+----------+
```

467. scott의 월급을 0으로 변경하세요~

```
mysql> update emp
    -> set sal = 0
    -> where ename = 'scott';
```

이건 다시 rollback 해도 적용 안된다.

❖ **mysql은 오라클과 다르게 기본 자동커밋이 활성화 되어있다.**
자동 커밋되어버려서 rollack을 할 수 없습니다.
mySQL 사용자들이 많이 하는 실수중에 하나이니 주의할것!!

468. 자동 커밋이 활성화 되어져 있는지 확인합니다.

```
select @@autocommit;
```

결과:

```
+--------------+
| @@autocommit |
+--------------+
|            1 |
+--------------+
```

설명: 숫자 1이면 autocommit이 켜 있는 것 입니다.

469. 자동 커밋 기능을 끄세요~

```
set autocommit = FALSE;
```

다시 확인해보면 :

```
mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            0 |
+--------------+
```

이렇게 되어있다.

470. KING의 월급을 0으로 변경하세요!

```
mysql> update emp
    -> set sal = 0
    -> where ename = 'KING';
```

결과:

```
| empno | Ename  | Job       | Mgr  | Hiredate   | Sal  | Comm | Deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 |    0 | NULL |     10
```

하고 rollback 해보면

```
-------+--------+-----------+------+------------+------+------+--------+
| empno | Ename  | Job       | Mgr  | Hiredate   | Sal  | Comm | Deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
```

다시 돌아가있다.

설명: mySQL을 켰으면 제일 먼저 확인해야하는 것이 자동커밋이 활성 되었는지 확인

하는것!!!!

471. 오라클의 listagg함수와 같은 기능인 group_concat을 이용해서 부서번호, 부서번호별로 속한 사원들의 이름을 가로로 출력하시오!

```
mysql> select deptno, group_concat(ename separator '/')
    -> from emp
    -> group by deptno;
```

결과:

```
+--------+----------------------------------+
| deptno | group_concat(ename separator '/') |
+--------+----------------------------------+
|     10 | KING/CLARK/MILLER                |
|     20 | JONES/FORD/SMITH/SCOTT/ADAMS     |
|     30 | BLAKE/MARTIN/TURNER/JAMES/WARD   |
+--------+----------------------------------+
```

472. mySQL과 파이썬을 연동해서 mySQL의 emp테이블을 파이썬에서 출력하시오!

먼저 아나콘다 프롬프트 창을 열고 pymysql을 설치하세요!

conda install pymysql

설치되면

스파이더에

```
import pymysql,pandas as pd  #임포트 할 때 ',' 로 연결해서 같이 붙일 수 있다.

conn = pymysql.connect(host="localhost", user="root",password="oracle",
db="orcl",charset="utf8")

curs = conn.cursor()
sql = "select * from emp"
curs.execute(sql)
rows = curs.fetchall()
colname = curs.description
col = []
for i in colname:
    col.append(i[0].upper())
emp = pd.DataFrame(list(rows),columns=col)
print(emp[['ENAME', 'SAL']] )
```

473. 직업, 직업별 토탈월급을 출력하시오!

```
import pymysql,pandas as pd

conn = pymysql.connect(host="localhost", user="root",password="oracle",
db="orcl",charset="utf8")
```

```
curs = conn.cursor()
sql = "select job, sum(sal) from emp group by job"
curs.execute(sql)
rows = curs.fetchall()
colname = curs.description
col = []
for i in colname:
    col.append(i[0].upper())
emp = pd.DataFrame(list(rows),columns=col)
print(emp)
```

474. 위의 결과를 막대그래프로 시각화 하시오!

475. (오늘의 마지막 문제 12/21)

오라클과 파이썬을 연동하여 아래의 사원들을 검색하시오!
이름과 월급, 부서번호와 자기가 속한 부서번호의 평균 월급을 출력하는데 자신의월급이 자기가 속한 부서번호의 평균 월급보다 더 큰 사원들만 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn("localhost", 1521, 'orcl')
db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select e.ename, e.sal, v.deptno, v.부서평균
                  from emp e,
                      (select deptno, avg(sal) 부서평균
                          from emp
                          group by deptno) v
                where e.deptno = v.deptno and e.sal> v.부서평균
                  """)
row = cursor.fetchall()

emp = pd.DataFrame(row)
print(emp)
```

476. 다음에서 출력되고 있는 리스트 안의 요소들은 문자입니다.ㅏ 근데 문자가 아니라 리스트의 요소들이 숫자가 되게 하세요~

답:

```
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
```

```
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것
('')인 null로 변경해라~
        file_name.append(int(a))
    return file_name

print(image_load(test_image))
```

477. 위의 리스트 요소가 ascending하게 정렬되게 하시오!

```
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것
('')인 null로 변경해라~
        file_name.append(int(a))
        file_name.sort()
    return file_name

print(image_load(test_image))
```

478. 위의 함수의 코드를 추가해서 아래와 같이 출력되게 하시오!

결과:

['1.png', '2.png', '3.png', '4.png', '5.png', '6.png', '7.png', '8.png', '9.png', '10.png', '11.png', '12.png', '13.png', '14.png', '15.png', '16.png', '17.png', '18.png', '19.png', '20.png']

```
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인것
('')인 null로 변경해라~
        file_name.append(int(a))
        file_name.sort()

    file_res = []
```

```
    for k in file_name:
        file_res.append(str(k)+'.png')
    return file_res

print(image_load(test_image))
```

479. 위 함수의 코드를수정해서 아래와 같이 이름 앞에 절대경로가 붙게하시오!

```
import os
import re #데이터 정제를 전문으로 하는 모듈

test_image = 'c:₩₩images'

def image_load(path):
    file_list = os.listdir(path) #해당 디렉토리의 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]','', i) #i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인
                                   #('')인 null로 변경해라~
        file_name.append(int(a))
        file_name.sort()

    file_res = []
    for k in file_name:
        file_res.append('c:₩₩images₩₩' + str(k)+'.png')
    return file_res

print(image_load(test_image))
```

480. (점심시간 문제) 지난번에 직접 스크롤링한 사진 20장을 c드라이브 밑에 images3에 넣고 숫자로 변환하는 함수를 image_load3로 생성하시오!

```
import  os
import  re
import  cv2
import  numpy  as np
path ="c:₩₩images3"

def  image_load3(path):
    file_list = os.listdir(path)
    file_name =[]
    for   i   in  file_list:
        a = re.sub('[^0-9]', '', i)
        file_name.append( int(a) )
        file_name.sort()

    file_res=[]
    for  k  in  file_name:
        file_res.append('c:₩₩images3₩₩' + str(k) +'.jpg')

    image =[]
    for  h  in  file_res:
```

```
        img = cv2.imread(h)
        image.append(img)

    return  np.array(image, dtype = object)


print (image_load3(path))
```

결과:

```
 [ 92,  29,  25],
        [ 94,  32,  26],
        [ 89,  36,  33]]]
 :     :       :
```

이런식으로 나옴

481. 아래의 두 행렬을 만들고 덧셈 연산을 하시오!

```
import numpy as np
a = [ [ 1,2,3], [0,1,2], [3,0,1] ]
b = [[2,0,1], [0,1,2], [1,0,2]]
a2 = np.array(a)
b2 = np.array(b)
print(a2 + b2)
```

482. 아래의 두 행렬을 만들고 두 행렬의 원소의 곱을 구하시오!

```
import numpy as np
a = [ [ 1,2,3], [0,1,2], [3,0,1] ]
b = [[2,0,1], [0,1,2], [1,0,2]]
a2 = np.array(a)
b2 = np.array(b)
print(a2 * b2)
```

결과:

```
[[2 0 3]
 [0 1 4]
 [3 0 2]]
```

483. 문제 482번에서 출력된 결과의 요소들을 모두 다 더하시오!

```
import numpy as np
a = [ [ 1,2,3], [0,1,2], [3,0,1] ]
b = [[2,0,1], [0,1,2], [1,0,2]]
a2 = np.array(a)
b2 = np.array(b)
result = a2*b2
print( np.sum(result)) #행렬 안의 원소들의 합을 출력합니다.
```

484. 아래의 4*4행렬을 만드시오!

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]
a2 = np.array(a)

print(a2)
```

결과:

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]
```

485. 아래의 4x4 행렬에서 빨간색으 지정한영역들의 숫자만 출력하시오!

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]
```

설명: 행 - 1~3까지
열 - 1~3까지

답:

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

print(a2[0:3, 0:3]) #행/열을 0이상에서 3미만까지
```

486. 아래 4x4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오!

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]
```

답:

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

print(a2[0:3, 1:4])
          행   열
```

487. 아래의 4x4행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오!

[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]

답:

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

print(a2[1:4, 0:3])
```

488. 아래의 4x4행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오!

[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]

답:

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

print(a2[1:4, 1:4])
```

489. 아래의 4x4행렬에서 위에서 빨간색으로 지정된 4개의 영역의 숫자들을 for loop문을 이용해서모두 출력하시오!

```
import numpy as np

a = [[1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1]]
a2 = np.array(a)

for i, k in zip(range(0,2), range(3,5)):
    for j, h in zip(range(0,2), range(3,5)):
        print(a2[i:k, j:h])
```

또는

```
import numpy as np
a = [ [ 1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

for i in range(0,2):
      for k in range(0,2):
          print(a2[i:i+3, k:k+3])
```

490. 문제 489번에서 선택한 4개의 행렬(3x3)과 아래의 filter 행렬(3x3)과의 원소의 곱을 출력하시오.

```
import numpy as np

f = [ [2,3,4],[1,2,3],[2,0,1] ]
filter = np.array(f)
print(filter)

import numpy as np
a = [ [1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

for i in range(0,2):
        for k in range(0,2):
            print(a2[i:i+3,k:k+3] * filter)
```

결과:

```
[[2 3 4]
 [1 2 3]
 [2 0 1]]
[[ 2  6 12]
 [ 0  2  6]
 [ 6  0  1]]
[[4 9 0]
 [1 4 9]
 [0 0 2]]
[[0 3 8]
 [3 0 3]
 [4 0 0]]
[[ 2  6 12]
 [ 0  2  6]
 [ 6  0  1]]
```

491. 위에서 출력된 3x3행렬 4개에 대한 원소들의 합이 각각 출력되게 하시오

```
import numpy as np

f = [ [2,3,4],[1,2,3],[2,0,1] ]
filter = np.array(f)

import numpy as np
a = [ [1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

for i in range(0,2):
        for k in range(0,2):
            print(np.sum(a2[i:i+3,k:k+3] * filter))
```

결과:

```
35
```

29
21
35

492. 위에서 출력된 4개의숫자 아래의행렬 2x2를 만드시오!

```
import numpy as np

f = [ [2,3,4],[1,2,3],[2,0,1] ]
filter = np.array(f)

import numpy as np
a = [ [1,2,3,0], [0,1,2,3], [3,0,1,2], [2,3,0,1] ]

a2 = np.array(a)

result = []
for i in range(0,2):
        for k in range(0,2):
            result.append(np.sum(a2[i:i+3,k:k+3] * filter))

result2 = np.array(result).reshape(2,2) #넘파이 어레이의 2x2 행렬로 변경
print(result2)
```

설명 : a2라는 원본이미지(개사진)에 filter(랜덤으로 생성한 이미지)를 가지고 원본이미지를 스트라이드(양옆, 위아래로 스캔) 하면서 특징을 잡아내어 특징이미지를 추출(result2)하는것을 합성곱이라고 합니다.

493. 아래의 원본 이미지 행렬(5x5)에서 필터행렬(4x4)로 스트라이딩(합성곱)해서 특징(2x2)을 추출하시오~

4x4 ---> 3x3 ---> 2x2
5x5 ---> 4x4 ---> 2x2

답:

```
import numpy as np

f = [ [3,1,4,1],[2,3,3,4],[5,1,2,1], [6,1,3,4] ]
filter = np.array(f)

import numpy as np
a = [ [2,3,1,4,7], [3,1,6,4,3], [2,1,5,3,1], [6,2,4,1,2], [7,3,1,2,3]]
a2 = np.array(a)

result = []
for i in range(0,2):
   for k in range(0,2):
                result.append(np.sum(a2[i:i+4,k:k+4] * filter))

result2 = np.array(result).reshape(2,2) #넘파이 어레이의 2x2 행렬로 변경
print(result2)
```

494. 아래의 리스트에서 숫자3이 있는지 숫자 탐색으로 구현하시오! 있으면 '숫자 3이 있습니다'. 라는 메세지가 출력되게 하시오

설명: 숫자 탐색이란? 주어진 데이터를 처음부터 차례대로 비교하면서 탐색하는 방법

```
a = [15,11,1,3,8]
```

```
a = [15,11,1,3,8]

for i in a:
    if i == 3:
        print('숫자 3이 있습니다')
        break
else:
    print('숫자 3이 없습니다')
```

495. 위의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자가 존재하는지 존재하지 않는지가 출력되게 하시오!

```
a = [15,11,1,3,8]

num = input('숫자를 입력하세요~')

for i in a:
    if i == num:
        print('숫자',num,'이 있습니다')
        break
else:
    print('숫자',num,'이 없습니다')
```

496. 아래의 a리스트에서 중앙값을 찾으시오!
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

답:

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

import numpy as np
a_n = np.array(a)
print(int(np.median(a_n)))
```

497. a리스트에서 첫번째 숫자부터 중앙값에 해당하는 숫자까지 검색하시오~

힌트: a.index(요소명)

답:

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

import numpy as np
a_n = np.array(a)
a_m = np.median(a_n)

print(a[:a.index(a_m)+1])
```

498. 위의 a리스트에서 문제 497번에서 선택된 숫자들의 중앙값까지 포함해서 지우고 아래의 결과 출력되게하시오!

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

import numpy as np
a_n = np.array(a)
a_m = np.median(a_n)

del(a[:a.index(a_m)+1])
print(a)
```

499. 위의 결과로 출력된 아래의 리스트에서 중앙값을 출력하시오!

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

import numpy as np
a_n = np.array(a)
a_m = np.median(a_n)

del(a[:a.index(a_m)+1])
a_m2 = np.median(a)
print(a_m2) #77.0
```

500. 위에서 출력한 중앙값 77이 내가 검색하고자하는 67보다 크다면 아래의 결과 리스트만 출력되게하시오

결과 : [51, 64, 67]

답:

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

import numpy as np
a_n = np.array(a)
a_m = np.median(a_n)

del(a[:a.index(a_m)+1])
a_m2 = np.median(a)

if a_m2 > 67:
    del(a[a.index(a_m2):])
else:
    del(a[:a.index(a_m2)+1])
```

```
print(a)
```

501. (오늘의 마지막 문제 12/22) EBS에 나온 영상대로 이진탐색을 구현하시오!

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
```

a 리스트에서 검색할 자를 입력하세요~ 67
67은 이진탐색 3번만에 검색되었습니다.

답:

```
import numpy as np
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

b = int(input('숫자를 입력하세요~'))

cnt = 0
while True:
        cnt +=1
        a_m2 = np.median(a)

        if np.median(a_m2) > b:
            del(a[a.index(a_m2):])

        elif np.median(a_m2) < b:
            del(a[:a.index(a_m2)+1])

        else:
            del(a[0:a.index(a_m2)])
            del(a[a.index(a_m2)+1:])
            break
print(b,'는 이진탐색', cnt,'번만에 검색되었습니다.')

if b not in a:
    print(b,'은 리스트에 없습니다')
```

이진탐색 과정을 보여주는 코드:

```
num = int(input('숫자를 입력하시오'))
import numpy as np
a = [ 1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871 ]
for i in range(1,5):
  a_n = np.array(a)  #num이 리스트 안에 있으면 아래의 실행문을 실행하겠다.
  a_m = np.median(a_n)
  if num in a: #67 --[67] 아래의 문장을 실행하면서 break 합니다.
    if num == a_m:
      print(num, '은 ', i, '번에 나왔습니다.')
      break
    elif num < a_m: #내가 검색하고자 하는 숫자가 중앙값보다 작다면
      del( a[ a.index(a_m)  : ] ) #중앙값부터 끝까지의 요소를 a리스트에서 지운다.
      print(a)
```

```
    else: #67이 47보다 크므로 아래의 문장이 실행이 됩니다.
      del( a[ :a.index(a_m)+1 ] )
      print(a)
  else: #그렇지 않다면 아래의 실행문으로 실행하겠다.
    print(num,'은 리스트에 없습니다.')
    break
```

502. 아래의 리스트를 만들고 첫번째 요소와 두번째 요소의 순서를 변경하시오!
a = [10, 5, 20, 9, 8]
결과: [5, 10, 20, 9, 8]

답:

```
a = [10, 5, 20, 9, 8]

temp = a[1] #1번째 요소를 temp변수에 임시저장
a[1] = a[0] #1번째 요소를 0번째 요소로 변경
a[0] = temp #0번째 요소를 1번째 요소로 변경합니다.
print(a)
```

503. 아래의 a리스트의 첫번째 요소와 두번째 요소의 크기를 비교해서 첫번째 요소의 숫자가 두번째 요소의 숫자보다 크다면 두개를 바꿔치기해라~
a = [10, 5, 20, 9, 8]
결과: [5, 10, 20, 9, 8]

답:

```
a = [10, 5, 20, 9, 8]


if a[0] > a[1]:
    temp = a[1] #1번째 요소를 temp변수에 임시저장
    a[1] = a[0] #1번째 요소를 0번째 요소로 변경
    a[0] = temp #0번째 요소를 1번째 요소로 변경합니다.
print(a)
```

504. 문제 503번 코드에 아래의 for loop문을 넣어서 버블정렬 하시오!
a = [5,4,3,2,1,8,7,10]
결과:
[4, 5, 3, 2, 1, 8, 7, 10]
↓
[4, 3, 5, 2, 1, 8, 7, 10]
↓
[4, 3, 2, 5, 1, 8, 7, 10]
↓
[4, 3, 2, 1, 5, 8, 7, 10]

↓

[4, 3, 2, 1, 5, 8, 7, 10]

↓

[4, 3, 2, 1, 5, 7, 8, 10]

```
a = [5,4,3,2,1,8,7,10]

for i in range(0,6):
    if a[i] > a[i+1]:
        temp = a[i+1]
        a[i+1] = a[i]
        a[i] = temp
    print(a)
```

505. (필수알고리즘3) 위의 코드를 이용해서 버블정렬을 하는 함수를 아래와 같이 생성하시오!

a = [5,4,3,2,1,8,7,10]
print(bubble_sort(a))

답:

```
a = [5,4,3,2,1,8,7,10]

def bubble_sort(a):
    for k in range(10):
        for i in range(len(a)-1):
            if a[i] > a[i+1]:
                temp = a[i+1]
                a[i+1] = a[i]
                a[i] = temp
    return a
print(bubble_sort(a))
```

또는

김소라씨 코드:

```
##############################################
a = [5, 4, 3, 2, 1, 8, 7, 10]
def bubble_sort(num):
    n = len(num)

    for k in range(n): #0-7
        cnt = 0 #실행문1
        for i in range(n-1):      #실행문2
            if num[i] > num[i+1]:
                temp = num[i+1]
                num[i+1] = num[i]
                num[i] = temp
                cnt = 1
```

```
        if cnt == 0:    #실행문3
            break
    return num

print(bubble_sort(a))
```

한결씨 코드:

```
a=[10,1,2,3,4,5,6,7,8,9]
def bubble_sort(a):
    n = 1
    j = -1 #한번 정렬한 숫자는 다시 계산 안하게 해주는 역할을 위해 j에 -1 할당
    while n != 0:
        n = 0
        j += 1
        for i in range(len(a)-j-1): # 과도한 계산을 막기 위해서 하나씩 빼줌 처음에 정렬된 제일 큰 수는
                                    # 안봐도 되니까
            if a[i] > a[i+1]:
                a[i],a[i+1] = a[i+1],a[i] #한번에 교환해버리게끔 구현
                n += 1  #교환할게 있으면 n은 증가합니다.
    return a

print(bubble_sort(a))
```

506. 14를 10으로 나눈 몫을 출력하시오!

```
print(14/10) #1.4
print(int(14/10)) #1
###############

print(14//10) #1
```

507. 14를 10으로 나눈 나머지값을 출력하시오!

```
print(14%10) #4
```

508. 숫자를 물어보게하고 숫자를 10으로 나눈 몫과 숫자를 10으로 나는 나머지 값을 출력하게 하시오!

```
num = int(input('숫자를 입력하세요~'))

print('몫은:',num//10)
print('나머지 값은:', num%10)
```

509. 아래의 잔돈 리스트를 만들고 잔돈 리스트의 첫번째 요소로 나눈몫과 나머지값이 출력

되게 하시오.
coin = [10, 7, 1]

#예시
숫자를 입력하세요~ 14
몫은 :1
나머지값은 : 4

답:
```
coin = [10, 7, 1]
num = int(input('숫자를 입력하세요~'))

print('몫은:',num//coin[0])
print('나머지 값은:', num%coin[0])
```

510. 위의 코드를 함수로 만들어서 실행되게 하시오!
greedy()
잔돈을 입력하세요~14
몫은: 1
나머지 값은: 4

답:
```
def greedy():
    coin = [10, 7, 1]
    num = int(input('잔돈을 입력하세요~'))

    print('몫은:',num//coin[0])
    print('나머지 값은:', num%coin[0])

greedy()
```

511. (필수 알고리즘 4번째) 탐욕 알고리즘을 파이썬으로 구현하시오. (점심시간 문제)
greedy ()
잔돈을입력하세요~
10원 1개, 7원 0개

답:

```
def greedy() :
    coin = [10,7,1]
    coin = sorted( coin, reverse = True) #큰 값을 맨 앞으로 오게끔 정렬
    num = int(input('숫자를 입력하시오 : '))
    a = []
```

```
    for i in range(3):
        result = num//coin[i]
        num = num%coin[i]
        a.append(result)
    print('10원 동전',a[0],'개, 7원 동전',a[1],'개, 1원 동전',a[2], '개로 줍니다.')
greedy()
```

512. factorial 함수를 만드시오.

print(factorial(5))

5 x 4 x 3 x 2 x 1 = 120

답:

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n *factorial(n-1)

print(factorial(5))
```

```
          ↓
          5 x factorial(4)
                ↓
                4 x factorial(3)
                      ↓
                      3 x factorial(2)
                            ↓
                            2 x factorial(1)
```

513. 위의 factorial 함수를 재귀를 이용하지 말고 for loop문으로 구현하시오!

```
def factorial(n):
    cnt = 1
    for i in range(1, n+1):
        cnt *= i
    return cnt

print(factorial(5)) #120
```

※ 재귀를 이용할 때 2가지 장점?

a. loop문을 복잡하게 이용하지 않아도 됩니다.

b. 코드가 더 간결해 집니다.

514. 오라클의 power함수를 파이썬으로 구현하시오!

SQL > select power(2,3) from dual; #8

print(power(2,3)

답:

```
def power(n1, n2):
    cnt = 1
    for i in range(1, n2 + 1):
        cnt = cnt*n1
    return cnt

print(power(2,3))
```

515. 위의 power함수를 loop문을 쓰지 말고 재귀함수로 구현하시오!
print(power(2,3)) #8

답:

```
def power(x, n):
    if n == 0:
        return 1
    else:
        return x*power(x, n-1)

print( power(2,3) )
```

```
                ↓
          2 * power(2,2)
              ↓
            2 * power(2,1)
                ↓
              2 * power(2,0)
                  ↓
                  1
```

516. 구구단 2단을 아래와 같이 출력하는함수를 생성하시오.
(for loop문을 사용해서 만드세요!)

```
print( multi_table_2dan(9))

2 x 1 = 2
2 x 2 = 4
  :
  :
2 x 9 = 18
```

답:

```
def multi_table_2dan(n):
    for i in range(1, n+1 ):
        print('2 x', i , '=', 2*i)

multi_table_2dan(9)
```

517. 두 숫자를 입력해서 함수를 실행하면 두 숫자의 최대공약수가 출력되는 함수를 생성하시오!

(재귀함수 사용하지 않고 loop문 이용해서 하세요)

print(gcd(16, 24))

답:

```
def gcd(n1, n2):
   for i in range(max(n1, n2), 0, -1):
      if n1%i == 0 and n2%i == 0:
         return i

print(gcd(8,16))
```

518. (필수 알고리즘 5번째) 최대공약수를 출력하는 함수를 재귀함수로 구현하시오!

```
def gcd(n1,n2):
    if n2 == 0:
        return n1  #
    else:
        return gcd(n2, n1%n2)

print(gcd(16,24))
```

519. (오늘의 마지막 문제 12/23) 구구단 2단을 재귀함수로 출력하시오!

```
def multi_table_2(n, k=2):
    print(k,'x', 10-n,'=', k*(10-n))
    if n == 1:
        return k
    else:
        return k+multi_table_2(n-1)

multi_table_2(9)
```

520. (점심시간 문제) 지금 만든 cacheProcess함수를 녹색 코드로 검증하세요!

```
def cacheProcess( cities, cachesize):
    city = [i.lower() for i in cities]
    cache = [None for i in range(1,5)]
    cnt = 0
    for i in range(0, len(city)):
        if city[i] in cache:
            cnt = cnt+1
            cache.append( city[i] )
            del cache[0]

        elif city[i] not in cache:
            cnt = cnt+5
            cache.append( city[i] )
            del cache[0]

    return cnt
```

```
cities = ['Jeju', 'Pangyo', 'Seoul', 'Jeju', 'Pangyo','Seoul', 'Jeju', 'Pangyo','Seoul']
print(cacheProcess(cities,3)) #21
cities = ['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo', 'Seoul', 'NewYork',
'LA']
print(cacheProcess(cities,3)) #50
cities = ['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'SanFrancisco', 'Seoul', 'Rome',
'Paris', 'Jeju', 'NewYork', 'Rome']
print(cacheProcess(cities,2)) #52
cities = ['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'SanFrancisco', 'Seoul', 'Rome',
'Paris', 'Jeju', 'NewYork', 'Rome']
print(cacheProcess(cities,5)) #52
cities = ['Jeju', 'Pangyo', 'NewYork', 'newyork']
print(cacheProcess(cities,2)) #16
cities = ['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA']
print(cacheProcess(cities,0)) #25
cities = ['Jeju', 'Jeju','Jeju']
print(cacheProcess(cities,3))
#%%
check = []

a = cacheProcess(["Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul", "Jeju",
"Pangyo", "Seoul"],3)

check.append(a)

a = cacheProcess(["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "Jeju", "Pangyo",
"Seoul", "NewYork", "LA"],3)

check.append(a)

a =
cacheProcess(["Jeju","Pangyo","Seoul","NewYork","LA","SanFrancisco","Seoul","Rome",
"Paris","Jeju","NewYork","Rome"],2)

check.append(a)

a =
cacheProcess(["Jeju","Pangyo","Seoul","NewYork","LA","SanFrancisco","Seoul","Rome",
"Paris","Jeju","NewYork","Rome"],5)

check.append(a)

a = cacheProcess(["Jeju","Pangyo",'NewYork','newyork'],2)

check.append(a)

a = cacheProcess(["Jeju","Pangyo","Seoul","NewYork","LA"],0)

check.append(a)

a = cacheProcess(['Jeju', 'Jeju','Jeju'],3)
```

```
check.append(a)


print(check)


correct = [21,50,52,52,16,25,7]


for i in range(len(check)) :
    if check[i] != correct[i] :
        print("%i번째 경우가 틀립니다."%i)
```

521. 아래의 코드를 함수로 만들어서 아래와 같이 실행되게 하시오!

(코드)

```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']

union = set(a + b)

intersection = []
for i in a:
    if i in b:
        intersection.append(i)

import math
len_i = len(intersection)
len_u = len(union)
print(math.trunc(len_i/len_u * 65536)) #16384

print( Jaccard( str1, str2 ) )
```

문자열을 입력해주세요. : FRANCE

문자열을 입력해주세요. : french

16384

답:

```
def  str_split(string):
    res =[]
    for  i  in  range( len(string) -1 ):
        if  string[i].isalpha()  and  string[i+1].isalpha():
            res.append( string[ i : i+2] )
    return res

def Jaccard():
    import  math
```

```
    str1 = input('문자열을 입력하세요 ~').upper()
    str2 = input('문자열을 입력하세요 ~').upper()

    a = str_split(str1)
    b = str_split(str2)

    intersection = []
    for  i  in  a:
        if  i  in  b:
            intersection.append(i)

    union = set( a + b )

    len_i = len(intersection)
    len_u = len(union)
    return (   math.trunc( len_i / len_u *   65536) )


print( Jaccard() )
```

522. 문제 521번까지에서에 만든 Jaccard함수에 아래의 내용을 코드에 추가하시오!
print(Jaccard())

문자열을 입력해주세요. : E = M*C^2
문자열을 입력해주세요. : e = m*c^2

a와 b가 모두 공집합일 경우에는 나눗셈이 정의되지 않으니 따로 J(A,B) = 1로 정의한다.

```
def  str_split(string):
    res =[]
    for  i  in  range( len(string) -1 ):
        if  string[i].isalpha()  and  string[i+1].isalpha():
            res.append( string[ i : i+2] )
    return res

def Jaccard():
    import  math

    str1 = input('문자열을 입력하세요 ~').upper()
    str2 = input('문자열을 입력하세요 ~').upper()

    a = str_split(str1)
    b = str_split(str2)

    intersection = []
    for  i  in  a:
        if  i  in  b:
            intersection.append(i)
```

```
    union = set(a+b)

    len_i = len(intersection)
    len_u = len(union)
    if len_i == 0 and len_u == 0:
        return 65536
    return (  math.trunc( len_i / len_u *  65536) )


print( Jaccard() )
```

523. set 이용하지 말고 아래의 A와 B 두개의 리스트의 합집합이 { 1,1,1,2,2,3,4,5 } 로 출력되게 하시오!

```
A = [1, 1, 1, 2, 2, 3]
B = [1, 1, 2, 2, 4, 5]
intersection=[]
for  i  in  A:
    if  i  in B:
        intersection.append(i)
print(intersection)   #[1, 1, 1, 2, 2]

s = A + B
print(s) #[1, 1, 1, 2, 2, 3, 1, 1, 2, 2, 4, 5]
diff = []
for  i  in  s:
    if  i  not  in  intersection:
        diff.append(i)
print(diff) #[3, 4, 5]

union = diff + intersection
print(union) #[3, 4, 5, 1, 1, 1, 2, 2]
```

위의 코드는 합집합은 맞지만 교집합이 이상하므로 한결씨가 구현 collections으로 구현하면 다음과 같습니다~ 맞는지 확인하세요!

```
import collections

a = [1,1,1,2,2,3]
b = [1,1,2,2,4,5]

intersection = []
result = collections.Counter(a) & collections.Counter(b) # 교집합
intersection = list(result.elements()) # 요소만 리스트로 빼내오기

result2 = collections.Counter(a) | collections.Counter(b) # 합집합
union = list(result2.elements())        # 요소만 리스트로 빼내오기
```

```
print(intersection)  #[1, 1, 2, 2]
print(union)  #[1, 1, 1, 2, 2, 3, 4, 5]
```

524. (오늘의 마지막문제 12/24) 한결씨가 알려준 collections를 이용해서 자카드유사도를 출력하세요!

문자열을 입력하세요~ handshake

문자열을 입력하세요~ shake hands

65536

문자열을 입력하세요~ FRANCE

문자열을 입력하세요~ french

16384

```
######################
답:
def  str_split(string):
    res =[]
    for  i  in  range( len(string) -1 ):
        if  string[i].isalpha()  and  string[i+1].isalpha():
            res.append( string[ i : i+2] )
    return res

def Jaccard():
    import  math

    str1 = input('문자열을 입력하세요 ~').upper()
    str2 = input('문자열을 입력하세요 ~').upper()

    a = str_split(str1)
    b = str_split(str2)

    result = collections.Counter(a) & collections.Counter(b) # 교집합
    intersection = list(result.elements()) # 요소만 리스트로 빼내오기

    result2 = collections.Counter(a) | collections.Counter(b) # 합집합
    union = list(result2.elements())       # 요소만 리스트로 빼내오기

    len_i = len(intersection)
    len_u = len(union)

    if len(a)==0 and len(b)==0:
        return 1 *65536

    return math.trunc( len_i / len_u *  65536)

print( Jaccard() )
```

525. 숫자 9를 이진수로 변환하시오! (목차 158)
결과 : 0b1001 #0b :이진 binary의 약자

```
print( bin(9) )
```

526. 문제 525번의 결과에서 0b는 안나오고 1001일만 출력되게하시오
결과 : 1001

```
print(bin(9)[2:])
```

527. 숫자 30의 이진수를 아래와 같이 출력하시ㅣ오
결과: 11110

```
print(bin(30)[2:])
```

528. 아래와 같이 출력되 change 함수를 생성하시오!

```
def change(num):
    return bin(num)[2:]


print(change(30))
```

529. 문제 528번에 만든 change 함수에 숫자 9를 넣고 출력해보시오!

```
print(change(9))
```

결과 : 1001

530. change함수를 실행할 때 아래와 같이 입 매개변수를 하나 더 만들어서 입력매개변수의 길이만 숫자 0이 채워져서 출력되게하시오!

```
def change(num, n):
    return bin(num)[2:].rjust(n, '0')

print(change(9,5))
```

결과:
01001

설명 : rjust(5, '0')은 출력되는 자리수를 전체 5자리로 잡고 나머지 왼쪽에 '0'을 채워넣어라~

531. 카카오 문제의 예제를 가지고 아래의 결과를 출력하시오!

```
n = 5
arr1 = [9, 20, 28, 18, 11]
arr2 = [30, 1, 21, 17, 28]
```

결과:
01001 11110

10100 00001
11100 10101
10010 10001
01011 11100

답:

```
def change(num, n):
    return bin(num)[2:].rjust(n, '0')

arr1 = [9, 20, 28, 18, 11]
arr2 = [30, 1, 21, 17, 28]
for i, k in zip(arr1, arr2):
    print(change(i, 5), change(k,5))

#%%


#또는
def change(num, n):
    return bin(num)[2:].rjust(n, '0')



n = 5
arr1 = [9, 20, 28, 18, 11]
arr2 = [30, 1, 21, 17, 28]

for i in range(len(arr1)):
        print( change(arr1[i], n), change(arr2[i], n) )
```

532. 문제 531번 문제에서 출력된 아래의 결과를 이용해서 두 지도의 숫자가 둘다 0이면 공백(" ")을 출력하고 아니면 벽('#')을 출력하시오!

```
def change(num, n):
        return bin(num)[2:].rjust(n, '0')



n = 5
arr1 = [9, 20, 28, 18, 11]
arr2 = [30, 1, 21, 17, 28]

result = []

for i in range(len(arr1)):
    a1 = change(arr1[i], n)
    a2 = change(arr2[i], n)
    f = ' '
    for k in range(n):
        if a1[k] == '0' and a2[k] == '0' :
            fi = ' '
            f = f + fi
        else:
            fi = '#'
            f = f+fi
```

```
    result.append(f)
print(result)
```

533. 문제 532번의 코드를 함수로 생성해서 아래와 같이 수행되게 하시오!

```
n = 6
arr1 = [46, 33, 33 ,22, 31, 50]
arr2 = [27 ,56, 19, 14, 14, 10]
출력   ["######", "### #", "## ##", " #### ", " #####", "### # "]
```

답:

```
def change(num, n):
    return bin(num)[2:].rjust(n, '0')



def secretmap(arr1, arr2, n):
    result = []

    for i in range(len(arr1)):
        a1 = change(arr1[i], n)
        a2 = change(arr2[i], n)
        f = ' '
        for k in range(n):
            if a1[k] == '0' and a2[k] == '0' :
                fi = ' '
                f = f + fi
            else:
                fi = '#'
                f = f + fi
        result.append(f)
    return result

n = 6
arr1 = [46, 33, 33 ,22, 31, 50]
arr2 = [27 ,56, 19, 14, 14, 10]

print(secretmap(arr1, arr2, n))
```

534. 동전 하나를 10000번 던져서 앞면이 나오는 횟수를 출력하시오!
(힌트 : random.randint(0,1))

답:

```
import random

cnt = 0
for i in range(1, 10001):
```

```
    result = random.randint(0,1) #숫자 0과 1중 하나를 랜덤으로 출력
    if result == 0: #0 앞면이라고 하겠습니다.
        cnt +=1

print(cnt)
```

535. 동전을 8번 던졌을 때 뒷면이 나오는 횟수를 출력하시오!

```
import random

cnt = 0
for i in range(1, 9):
    result = random.randint(0,1) # 숫자 0과 1중 하나를 랜덤으로 출력
    if result == 0: #0을 뒷면이라고 하겠습니다.
        cnt +=1

print(cnt)
```

536. 동전하나를 8번 던졌을 때 그 중 뒷면이 5번 나오는 횟수를 출력하시오!
(동전 하나를 8번 던지는 실험을 10000번 하세요)

```
import random

count = 0
for k in range(1, 10001):
    cnt = 0
    for i in range(1, 9):
        result = random.randint(0,1) # 숫자 0과 1중 하나를 랜덤으로 출력
        if result == 0: #0을 뒷면이라고 하겠습니다.
            cnt +=1

    if cnt == 5:
        count +=1

print(count)
```

537. 아래의 결과를 출력하시오.

```
print(a)
print(b)
```

결과:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8]
[0, 0, 0, 0, 0, 0, 0, 0, 0]
```

답:

```
a = [i for i in range(9)]
```

```
b = [0]*9

print(a)
print(b)
```

538. 동전 8번 던졌을 때 뒷면이 0번~8번 나오는횟수가 아래와 같이 출력되게 하시오!
(점심시간 문제)

결과 : [412, 3026, 10798, 22010, 27651, 21682, 10851, 3191, 0]

답:

```
import random

a = [i for i in range(9)]
b = [0]*9

for j in a:
    count = 0
    for k in range(1, 100001):
        cnt = 0
        for i in range(1, 9):
            result = random.randint(0,1) # 숫자 0과 1중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 하겠습니다.
                cnt +=1

        if cnt == j:
            count +=1
    b.append(count)
    del b[0]

print(b)
```

김홍비씨 코드:

```
import random
a = [ ]

for k in range(0,9):
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9):
            result = random.randint(0,1)  #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0:   #0을 뒷면이라고 가정
                cnt+=1
        if cnt==k:
            count+=1
    a.append(count)
print(a)
```

539. 위의 횟수를 가지고 아래와 같이 확률이 출력되게 하시오!
(횟수 / 100000)

```
import random
a = [ ]

for k in range(0,9):
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9):
            result = random.randint(0,1)  #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0:   #0을 뒷면이라고 가정
                cnt+=1
        if cnt==k:
            count+=1
    a.append(count)

b = [i/100000 for i in a]
print(b)
```

540. 위의 coin_cnt를 x축으로 하고 확률 b리스트를 y축으로 해서 막대그래프를 그리시오!

```
import random
import matplotlib.pyplot as plt


a = [ ]

for k in range(0,9):
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9):
            result = random.randint(0,1)  #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0:   #0을 뒷면이라고 가정
                cnt+=1
        if cnt==k:
            count+=1
    a.append(count)

b = [i/100000 for i in a]
coin_cnt = [0,1,2,3,4,5,6,7,8]

plt.bar(coin_cnt, b, tick_label = coin_cnt, align='center', color = 'red')
plt.show()
```

541. coin_cnt를 입력값으로 해서 coin_cnt의 요소들의 평균값과 표준편차 출력하시오!

```
import numpy as np

coin_cnt = [0,1,2,3,4,5,6,7,8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)

print(c_m) #4.0
print(c_s)  #2.581988897471611
```

542. 위에서 구한 평균값과 표준편차를 이용해서 coin_cnt의 요소에 대한 확률밀도함수 값을 출력하시오! (scipy.stats의 norm 사용)

norm.pdf(입력값, 평균값, 표준편차)

```
import numpy as np
from scipy.stats import norm

coin_cnt = [0,1,2,3,4,5,6,7,8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)

result = norm.pdf(coin_cnt, c_m, c_s)
print(result) #[0.04653742 0.0786696  0.11446359 0.14334535 0.15450968
0.14334535 0.11446359 0.0786696  0.04653742]
```

543. coin_cnt를 x축으로 두고 위의 확률밀도 함수값을 y축으로두어서 확률밀도 함수그래프를 그리시오

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

coin_cnt = [0,1,2,3,4,5,6,7,8]
c_m = np.mean(coin_cnt)
```

```
c_s = np.std(coin_cnt)

result = norm.pdf(coin_cnt, c_m, c_s)

plt.plot(coin_cnt, result, color = 'black')
plt.show()
```

544. 위의 그래프에서 신뢰구간 68%에 해당하는 부분만 녹색으로 색칠하시오!

- 신뢰구간범위
  a. 68% --> 평균 $\pm 1 *$ 표준편차
  b. 95% --> 평균 $\pm 1.96 *$ 표준편차
  c. 99% --> 평균 $\pm 2.58 *$ 표준편차

답:

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

coin_cnt = [0,1,2,3,4,5,6,7,8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)

result = norm.pdf(coin_cnt, c_m, c_s)

plt.plot(coin_cnt, result, color = 'black')
plt.fill_between(coin_cnt, result, where = (coin_cnt < c_m + c_s) & (coin_cnt >= c_m-c_s), color = 'green')
plt.show()
```

545. (파이썬 마지막문제) 위의 코드들을 이용해서 아래 같이 출력되게 하시오! - 도박사 문제

동전을 8번 던졌을 때 동전이 뒷면이 나오는 횟수가 신뢰구간 68% 안에 드는지 아닌

지 출력하는 함수를 생성하시오

print(coin_hypo(동전 뒷면의 횟수))
print(coin_hypo(4))

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 4번이 나올 확률이 신뢰구간 68% 안에 있습니다.

print(coin_hypo(8))
동전을 8번 던졌을 때 뒷면이 나오는 수가 8번이 나올 확률은 신뢰구간 68% 안에 없습니다.

```
import random
import numpy as np

def coin_avg_std(num):                    #동전을 몇번 던지냐에 따른 평균과 표준편차를 생성
    result=[]
    for i in range(num):
        cnt = 0
        for t in range(8):
            cnt += (random.randint(0,1))
        result.append(cnt)
    c_m = np.mean(result)
    c_s = np.std(result)
    return c_m, c_s


def coin_hypo(num):
    c_m,c_s = coin_avg_std(10000)        # 동전 몇번 던진지를 선택
    if c_m-c_s<=num<=c_m+c_s:
        return f'동전을  8번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간 68%안에 있습니다.'
    else:
        return f'동전을  8번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간 68%안에 없습니다.'

print(coin_hypo(4))
print(coin_hypo(8))
```